

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		1

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
while (i < 100) {  
    i++;  
}
```

kde i je proměná typu int proběhne:

**A** vždy 100x  
**B** alespoň jednou  
**C** kód obsahuje syntaktickou chybu  
**D** cyklus nemusí proběhnout ani jednou
- 2** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}
```

bude obsah proměné A roven:

**A** 97  
**B** 100  
**C** kód obsahuje syntaktickou chybu  
**D** 101  
**E** 98
- 3** Cyklus

```
int i = 0;  
do {  
    i++;  
} while (i < 100);
```

proběhne:

**A** 100x  
**B** kód obsahuje syntaktickou chybu  
**C** 99x  
**D** 101x
- 4** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x += 1; }
```

**A** 0  
**B** 1  
**C** 99  
**D** 100  
**E** kód se nepřeloží (obsahuje chybu)
- 5** Následující zápis reprezentuje:

```
double x[10];
```

**A** Pole s 10 prvky s indexy 1..10 typu double  
**B** Pole s 9 prvky s indexy 1..9 typu double  
**C** Pole s 9 prvky s indexy 0..9 typu double  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** kód se nepřeloží (obsahuje chybu)
- 6** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {  
}
```

proběhne:

**A** 99x  
**B** 100x  
**C** 101x  
**D** 0x  
**E** kód se nepřeloží (obsahuje chybu)
- 7** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;
```

Bude obsah proměné A roven:

**A** 300  
**B** kód obsahuje syntaktickou chybu  
**C** 100  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 200
- 8** Datový typ float je:

**A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání celých čísel
- 9** Výsledkem výrazu (1 && 0) je:

**A** 0  
**B** 10  
**C** není definováno  
**D** 1
- 10** Hodnota výrazu (3/2)\*(10+10)\*2 je:

**A** 0  
**B** 30  
**C** 40  
**D** 60  
**E** 4
- 11** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

**A** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 12** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 100
  - C** 301
  - D** 300
  - E** kód obsahuje syntaktickou chybu
  - F** 200
- 
- 13** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** kód se nepřeloží (obsahuje chybu)
  - B** " , Hello World!!"
  - C** "!"
  - D** "Hello World! , Hello World!"
  - E** "Hello, Hello World!!"
  - F** "Hello World!"
- 
- 14** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i < 3; i++)`  
`y = y + x[i];`
- A** 5
  - B** 3
  - C** 6
  - D** 9
  - E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		2

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

**A** 99x  
**B** 100x  
**C** 101x  
**D** kód obsahuje syntaktickou chybu
- 2** Hodnota výrazu  $(2+3)/5+5$  je:

**A** 6  
**B** 0  
**C** 10  
**D** 2  
**E** 1
- 3** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:

**A** není definováno  
**B** 0  
**C** 0 pokud je A rovno 0  
**D** 1
- 4** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 9  
**B** 5  
**C** 6  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 3
- 5** Následující zápis reprezentuje:

```
double x[10];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 9 prvky s indexy 1..9 typu double  
**C** Pole s 10 prvky s indexy 0..9 typu double  
**D** Pole s 10 prvky s indexy 1..10 typu double  
**E** Pole s 9 prvky s indexy 0..9 typu double
- 6** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

**A** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 8 znacích a textový řetězec o délce 8 znaků
- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** " , Hello World!!"  
**B** "!"  
**C** "Hello World!"  
**D** "Hello World! , Hello World!"  
**E** "Hello, Hello World!!"  
**F** kód se nepřeloží (obsahuje chybu)
- 8** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 102  
**D** 101
- 9** Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

**A** dvakrát  
**B** kód obsahuje syntaktickou chybu  
**C** ani jednou  
**D** jednou
- 10** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

**A** kód se nepřeloží (obsahuje chybu)  
**B** 100x  
**C** 101x  
**D** 99x  
**E** 0x
- 11** Datový typ int je:

**A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**C** Typ se znaménkem pro uchovávání celých čísel  
**D** Typ bez znaménka pro uchovávání celých čísel

- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = pA + 200;`  
Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** kód obsahuje syntaktickou chybu
  - C** 100
  - D** 300
  - E** 200
- 
- 13** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x += 1; }`
- A** 1
  - B** 99
  - C** 100
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 0
- 
- 14** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 301
  - B** 101
  - C** 100
  - D** 200
  - E** 300
  - F** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		3

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 9
- B kód se nepřeloží (obsahuje chybu)
- C 5
- D 6
- E 3

**2** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 30
- B 0
- C 4
- D 60
- E 40

**3** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:

- A 0
- B není definováno
- C 1
- D 10

**4** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
proběhne:
```

- A 0x
- B 101x
- C kód se nepřeloží (obsahuje chybu)
- D 99x
- E 100x

**5** Datový typ float je:

- A Typ se znaménkem pro uchovávání celých čísel
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ bez znaménka pro uchovávání celých čísel

**6** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:
```

- A 99x
- B 101x
- C kód obsahuje syntaktickou chybu
- D 100x

**7** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;
```

Bude obsah proměně A roven:

- A 300
- B 200
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D kód obsahuje syntaktickou chybu
- E 100

**8** Po provedení kódu:  

```
int A = 100;  
A = ++A + 200;
```

Bude obsah proměně A roven:

- A 200
- B 101
- C 100
- D 300
- E kód obsahuje syntaktickou chybu
- F 301

**9** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
    { x -= 1; }
```

- A -99
- B kód se nepřeloží (obsahuje chybu)
- C 1
- D -100
- E 0

**10** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "!"
- B "Hello World! , Hello World!!"
- C kód se nepřeloží (obsahuje chybu)
- D "Hello, Hello World!!"
- E "Hello World! , Hello World!"
- F "Hello World!"

- 11** Cyklus  
`int i = 100;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** dvakrát  
**C** ani jednou  
**D** jednou

- 12** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
    `default: A--;`  
`}`  
bude obsah proměné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 101  
**D** 98  
**E** 97

- 13** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků

- 14** Následující zápis reprezentuje:  
`int x[10];`  
**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 0..9 typu int  
**D** Pole s 9 prvky s indexy 1..9 typu int  
**E** Pole s 9 prvky s indexy 0..9 typu int

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		4

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
proběhne:
```

  
**A** 0x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 101x  
**D** 100x  
**E** 99x
- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "Hello World!"  
**B** " , Hello World!!"  
**C** "!"  
**D** "Hello World! , Hello World!"  
**E** "Hello, Hello World!"  
**F** kód se nepřeloží (obsahuje chybu)
- 3** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

  
**A** 9  
**B** 3  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 6  
**E** 5
- 4** Cyklus  

```
do {  
    i++;  
} while (i < 100);
```

  
kde i je proměnná typu int proběhne:  
  
**A** alespoň jednou  
**B** cyklus nemusí proběhnout ani jednou  
**C** kód obsahuje syntaktickou chybu  
**D** vždy 100x
- 5** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

  
**A** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 8 znacích a textový řetězec o délce 8 znaků
- 6** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:  
  
**A** není definováno  
**B** 0  
**C** 0 pokud je A rovno 0  
**D** 1
- 7** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;
```

  
} bude obsah proměnné A roven:  
  
**A** 98  
**B** 101  
**C** 97  
**D** 100  
**E** kód obsahuje syntaktickou chybu
- 8** Hodnota výrazu  $(2-20)/(5+5)$  je:  
  
**A** -1  
**B** 2  
**C** -1.8  
**D** 1  
**E** 0
- 9** Datový typ int má vždy velikost:  
  
**A** 2 bajty  
**B** v závislosti na architektuře  
**C** 8 bajty  
**D** 4 bajty
- 10** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;
```

  
Bude obsah proměnné A roven:  
  
**A** 100  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 200

- 11** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** 301
  - B** 300
  - C** 100
  - D** kód obsahuje syntaktickou chybu
  - E** 101
  - F** 200
- 
- 12** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x += 1; }`
- A** 99
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 1
  - D** 0
  - E** 100
- 
- 13** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 10 prvky s indexy 0..9 typu double
  - B** Pole s 9 prvky s indexy 0..9 typu double
  - C** Pole s 10 prvky s indexy 1..10 typu double
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole s 9 prvky s indexy 1..9 typu double
- 
- 14** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
    `i++;`  
`}`  
proběhne:
- A** 99x
  - B** 101x
  - C** 100x
  - D** kód obsahuje syntaktickou chybu



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		5

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Výsledkem výrazu (1 && 1) je:

- A 0
- B není definováno
- C 1
- D 10

**2** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```

bude obsah proměné A roven:

- A kód obsahuje syntaktickou chybu
- B 100
- C 98
- D 101
- E 97

**3** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

- A 100x
- B kód se nepřeloží (obsahuje chybu)
- C 0x
- D 99x
- E 101x

**4** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 9
- B 6
- C kód se nepřeloží (obsahuje chybu)
- D 5
- E 3

**5** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

- A 99
- B 100
- C 1
- D kód se nepřeloží (obsahuje chybu)
- E 0

**6** Datový typ char je:

- A Typ pro uchovávání velkých celých čísel
- B Typ pro uchovávání malých celých čísel nebo znaků
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ se znaménkem pro uchovávání čísel s desetinou čárkou

**7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A " , Hello World!!"
- B kód se nepřeloží (obsahuje chybu)
- C "!"
- D "Hello World! , Hello World!"
- E "Hello, Hello World!!"
- F "Hello World!"

**8** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
```

Bude obsah proměné A roven:

- A 100
- B 300
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 200
- E kód obsahuje syntaktickou chybu

**9** Hodnota výrazu (2+3)/5+5 je:

- A 2
- B 6
- C 1
- D 10
- E 0

**10** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A 99x
- B 100x
- C kód obsahuje syntaktickou chybu
- D 101x

**11** Cyklus  
`int i = 0;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:

- A** 100x
- B** kód obsahuje syntaktickou chybu
- C** 101x
- D** 99x

**12** Následující zápis reprezentuje:  
`int x[10];`

- A** Pole s 10 prvky s indexy 1..10 typu int
- B** Pole s 9 prvky s indexy 0..9 typu int
- C** Pole s 9 prvky s indexy 1..9 typu int
- D** Pole s 10 prvky s indexy 0..9 typu int
- E** kód se nepřeloží (obsahuje chybu)

**13** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`

- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
- B** Pole o 7 znacích a textový řetězec o délce 7 znaků
- C** Pole o 7 znacích a textový řetězec o délce 8 znaků
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**14** Po provedení kódu:

`int A = 100;`  
`A = --A + 200;`  
Bude obsah proměnné A roven:

- A** 100
- B** 101
- C** kód obsahuje syntaktickou chybu
- D** 200
- E** 299
- F** 300

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		6

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
```

proběhne:

  - A 100x
  - B 99x
  - C 101x
  - D kód obsahuje syntaktickou chybu
- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!!");
```

  - A "Hello World! , Hello World!!"
  - B "Hello World!"
  - C "!"
  - D kód se nepřeloží (obsahuje chybu)
  - E "Hello World! , Hello World!"
  - F "Hello, Hello World!!"
- 3** Následující zápis reprezentuje:

```
double x[10];
```

  - A Pole s 10 prvky s indexy 1..10 typu double
  - B Pole s 10 prvky s indexy 0..9 typu double
  - C Pole s 9 prvky s indexy 0..9 typu double
  - D kód se nepřeloží (obsahuje chybu)
  - E Pole s 9 prvky s indexy 1..9 typu double
- 4** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
```

Bude obsah proměnné A roven:

  - A 100
  - B kód obsahuje syntaktickou chybu
  - C 200
  - D 299
  - E 101
  - F 300
- 5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

  - A 9
  - B kód se nepřeloží (obsahuje chybu)
  - C 5
  - D 3
  - E 6
- 6** Datový typ char je:

  - A Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - B Typ pro uchovávání velkých celých čísel
  - C Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - D Typ pro uchovávání malých celých čísel nebo znaků
- 7** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

  - A kód se nepřeloží (obsahuje chybu)
  - B 99x
  - C 0x
  - D 100x
  - E 101x
- 8** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

  - A 66
  - B 0
  - C 33
  - D 1
  - E 44
- 9** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

  - A Pole o 8 znacích a textový řetězec o délce 7 znaků
  - B Pole o 7 znacích a textový řetězec o délce 8 znaků
  - C Pole o 7 znacích a textový řetězec o délce 7 znaků
  - D kód se nepřeloží (obsahuje chybu)
  - E Pole o 8 znacích a textový řetězec o délce 8 znaků
- 10** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

  - A 101x
  - B kód obsahuje syntaktickou chybu
  - C 100x
  - D 99x

- 11** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = &pA + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** 100
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 200
  - E** kód obsahuje syntaktickou chybu
- 
- 12** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
`}`  
bude obsah proměnné A roven:
- A** 100
  - B** kód obsahuje syntaktickou chybu
  - C** 101
  - D** 102
- 
- 13** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`if (1)`  
    `x = x + 1;`  
`else`  
    `x = x - 1;`  
    `for (int i = 0; i < 100; i++) {x++};`
- A** 0
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 1
  - D** 101
  - E** -1
- 
- 14** Výsledkem výrazu `(1 || A)`, kde A je proměnná typu integer je:
- A** 0 pokud je A rovno 0
  - B** 0
  - C** není definováno
  - D** nenulová hodnota

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		7

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Datový typ int je:

- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B** Typ bez znaménka pro uchovávání celých čísel
- C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D** Typ se znaménkem pro uchovávání celých čísel

**2** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A** nenulová hodnota
- B** 0
- C** 0 pokud je A rovno 0
- D** není definováno

**3** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

kde i je proměnná typu int proběhne:

- A** kód obsahuje syntaktickou chybu
- B** cyklus nemusí proběhnout ani jednou
- C** vždy 100x
- D** alespoň jednou

**4** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;  
Bude obsah proměnné A roven:
```

- A** 200
- B** obsah hodnoty A nelze bez spuštění jednoznačně určit
- C** 100
- D** kód obsahuje syntaktickou chybu
- E** 300

**5** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A** 0
- B** 60
- C** 40
- D** 30
- E** 4

**6** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
}
```

proběhne:

- A** 0x
- B** 99x
- C** 101x
- D** kód se nepřeloží (obsahuje chybu)
- E** 100x

**7** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu int
- B** Pole s 9 prvky s indexy 0..9 typu int
- C** Pole s 10 prvky s indexy 1..10 typu int
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole s 9 prvky s indexy 1..9 typu int

**8** Cyklus

```
int i = 99;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A** jednou
- B** kód obsahuje syntaktickou chybu
- C** dvakrát
- D** ani jednou

**9** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 6
- C** 9
- D** 3
- E** 5

**10** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 100
- C** 102
- D** 101
- E** 98

**11** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```

- A -100
- B -99
- C kód se nepřeloží (obsahuje chybu)
- D 0
- E 1

**12** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A "Hello World! , Hello World!"
- B "Hello, Hello World!!"
- C "!"
- D kód se nepřeloží (obsahuje chybu)
- E "Hello World!"
- F "Hello World! , Hello World!!"

**13** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 5 znacích a textový řetězec o délce 5 znaků
- B Pole o 6 znacích a textový řetězec o délce 5 znaků
- C Pole o 6 znacích a textový řetězec o délce 6 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 5 znacích a textový řetězec o délce 6 znaků

**14** Po provedení kódu:

```
int A = 100;
```

```
A = A++ + 200;
```

Bude obsah proměnné A roven:

- A 300
- B 101
- C 100
- D 200
- E 301
- F kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		8

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "!"
- B "Hello World!"
- C "Hello World! , Hello World!!"
- D "Hello, Hello World!!"
- E kód se nepřeloží (obsahuje chybu)
- F "Hello World! , Hello World!"

**2** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x += 1; }
```

- A 0
- B kód se nepřeloží (obsahuje chybu)
- C 100
- D 1
- E 99

**3** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
}
```

  
proběhne:

- A 101x
- B 0x
- C kód se nepřeloží (obsahuje chybu)
- D 100x
- E 99x

**4** Po provedení kódu:  

```
int A = 100;  
A = ++A + 200;
```

  
Bude obsah proměnné A roven:

- A 301
- B kód obsahuje syntaktickou chybu
- C 100
- D 300
- E 101
- F 200

**5** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}
```

  
proběhne:

- A kód obsahuje syntaktickou chybu
- B 100x
- C 101x
- D 99x

**6** Datový typ float je:

- A Typ bez znaménka pro uchovávání celých čísel
- B Typ se znaménkem pro uchovávání celých čísel
- C Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- D Typ bez znaménka pro uchovávání čísel s desetinnou čárkou

**7** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A 9
- B 6
- C kód se nepřeloží (obsahuje chybu)
- D 3
- E 5

**8** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

  
Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C 200
- D 300
- E 100

**9** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 44
- B 1
- C 66
- D 0
- E 33

**10** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A** 102
- B** 101
- C** 103
- D** kód obsahuje syntaktickou chybu

**11** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměnná typu int proběhne:

- A** cyklus nemusí proběhnout ani jednou
- B** vždy 100x
- C** kód obsahuje syntaktickou chybu
- D** alespoň jednou

**12** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 7 znaků
- B** Pole o 8 znacích a textový řetězec o délce 8 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 7 znacích a textový řetězec o délce 8 znaků
- E** Pole o 7 znacích a textový řetězec o délce 7 znaků

**13** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

- A** 0 pokud je A rovno 0
- B** 0
- C** není definováno
- D** 1

**14** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 1..9 typu int
- B** Pole s 10 prvky s indexy 0..9 typu int
- C** Pole s 10 prvky s indexy 1..10 typu int
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole s 9 prvky s indexy 0..9 typu int



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		9

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

- A 1
- B 0
- C 99
- D 100
- E kód se nepřeloží (obsahuje chybu)

**2** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměnné A roven:

- A 301
- B kód obsahuje syntaktickou chybu
- C 101
- D 100
- E 300
- F 200

**3** Datový typ int je:

- A Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- B Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
- C Typ bez znaménka pro uchovávání celých čísel
- D Typ se znaménkem pro uchovávání celých čísel

**4** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {
}
```

proběhne:

- A kód se nepřeloží (obsahuje chybu)
- B 0x
- C 101x
- D 100x
- E 99x

**5** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A 102
- B kód obsahuje syntaktickou chybu
- C 100
- D 98
- E 101

**6** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 4
- B 40
- C 30
- D 60
- E 0

**7** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A není definováno
- B 1
- C 0
- D 10

**8** Cyklus

```
int i = 100;
do {
    i++;
} while (i < 100);
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B dvakrát
- C jednou
- D ani jednou

**9** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 5 znacích a textový řetězec o délce 5 znaků
- B Pole o 6 znacích a textový řetězec o délce 6 znaků
- C Pole o 5 znacích a textový řetězec o délce 6 znaků
- D Pole o 6 znacích a textový řetězec o délce 5 znaků
- E kód se nepřeloží (obsahuje chybu)

**10** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = pA + 200;
```

Bude obsah proměnné A roven:

- A 200
- B 300
- C kód obsahuje syntaktickou chybu
- D 100
- E obsah hodnoty A nelze bez spuštění jednoznačně určit

**11** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměnná typu int proběhne:

- A kód obsahuje syntaktickou chybu
- B vždy 100x
- C cyklus nemusí proběhnout ani jednou
- D alespoň jednou

**12** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** "Hello World!"
- B** kód se nepřeloží (obsahuje chybu)
- C** "Hello, Hello World!!"
- D** "!"
- E** " , Hello World!!"
- F** "Hello World! , Hello World!"

**13** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A** 6
- B** 3
- C** 9
- D** 5
- E** kód se nepřeloží (obsahuje chybu)

**14** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 0..9 typu int
- B** Pole s 10 prvky s indexy 1..10 typu int
- C** Pole s 10 prvky s indexy 0..9 typu int
- D** Pole s 9 prvky s indexy 1..9 typu int
- E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		10

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Cyklus

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A 101x
- B kód obsahuje syntaktickou chybu
- C 99x
- D 100x

**2** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 300
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 100
- E 200

**3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A 9
- B kód se nepřeloží (obsahuje chybu)
- C 5
- D 3
- E 6

**4** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "!"
- C " , Hello World!!"
- D "Hello World! , Hello World!"
- E "Hello, Hello World!"
- F "Hello World!"

**5** Datový typ char je:

- A Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ pro uchovávání malých celých čísel nebo znaků
- D Typ pro uchovávání velkých celých čísel

**6** Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

- A jednou
- B ani jednou
- C kód obsahuje syntaktickou chybu
- D dvakrát

**7** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A 101
- B 100
- C 102
- D kód obsahuje syntaktickou chybu

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

- A 0
- B 1
- C -99
- D -100
- E kód se nepřeloží (obsahuje chybu)

**9** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {
}
```

proběhne:

- A 100x
- B 101x
- C 0x
- D kód se nepřeloží (obsahuje chybu)
- E 99x

- 10** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:
- A** 0 pokud je A rovno 0
  - B** 0
  - C** 1
  - D** není definováno
- 
- 11** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 100
  - C** 301
  - D** kód obsahuje syntaktickou chybu
  - E** 200
  - F** 300
- 
- 12** Hodnota výrazu  $(3/2)*(10+10)*2$  je:
- A** 4
  - B** 0
  - C** 30
  - D** 40
  - E** 60
- 
- 13** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - E** Pole o 6 znacích a textový řetězec o délce 6 znaků
- 
- 14** Následující zápis reprezentuje:  
`double x[10];`
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole s 9 prvky s indexy 1..9 typu double
  - C** Pole s 9 prvky s indexy 0..9 typu double
  - D** Pole s 10 prvky s indexy 0..9 typu double
  - E** Pole s 10 prvky s indexy 1..10 typu double

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		11

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  

```
int i = 100;  
do {  
    i++;  
} while (i < 100);
```

  
proběhne:  
**A** ani jednou  
**B** dvakrát  
**C** kód obsahuje syntaktickou chybu  
**D** jednou

- 2** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;
```

  
bude obsah proměnné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 101  
**C** 98  
**D** 100  
**E** 97

- 3** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

  
kde i je proměnná typu int proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** cyklus nemusí proběhnout ani jednou  
**C** vždy 100x  
**D** alespoň jednou

- 4** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;
```

  
Bude obsah proměnné A roven:  
**A** 100  
**B** 200  
**C** kód obsahuje syntaktickou chybu  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 300

- 5** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

  
**A** 1  
**B** -1  
**C** 101  
**D** 0  
**E** kód se nepřeloží (obsahuje chybu)

- 6** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
}
```

  
proběhne:  
**A** 99x  
**B** 100x  
**C** 0x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101x

- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "!"  
**B** "Hello World!"  
**C** " , Hello World!!"  
**D** "Hello, Hello World!!"  
**E** "Hello World! , Hello World!"  
**F** kód se nepřeloží (obsahuje chybu)

- 8** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 6  
**B** 2  
**C** 10  
**D** 1  
**E** 0

- 9** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:  
**A** 10  
**B** 0  
**C** není definováno  
**D** 1

**10** Datový typ `int` má vždy velikost:

- A 2 bajty
- B v závislosti na architektuře
- C 8 bajty
- D 4 bajty

**11** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 8 znaků
- B Pole o 8 znacích a textový řetězec o délce 7 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 8 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 7 znaků

**12** Následující zápis reprezentuje:

```
int x[10];
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole s 9 prvky s indexy 1..9 typu `int`
- C Pole s 10 prvky s indexy 1..10 typu `int`
- D Pole s 9 prvky s indexy 0..9 typu `int`
- E Pole s 10 prvky s indexy 0..9 typu `int`

**13** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A 9
- B 5
- C 3
- D 6
- E kód se nepřeloží (obsahuje chybu)

**14** Po provedení kódu:

```
int A = 100;  
A = ++A + 200;  
Bude obsah proměnné A roven:
```

- A 300
- B kód obsahuje syntaktickou chybu
- C 101
- D 100
- E 301
- F 200

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		12

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

**A** ani jednou  
**B** dvakrát  
**C** jednou  
**D** kód obsahuje syntaktickou chybu
- 2** Následující zápis reprezentuje:

```
double x[10];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 10 prvky s indexy 1..10 typu double  
**C** Pole s 9 prvky s indexy 0..9 typu double  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** Pole s 9 prvky s indexy 1..9 typu double
- 3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 5  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 6  
**D** 3  
**E** 9
- 4** Datový typ char je:

**A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**B** Typ pro uchovávání malých celých čísel nebo znaků  
**C** Typ pro uchovávání velkých celých čísel  
**D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- 5** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
{ x += 1; }
```

**A** 1  
**B** 0  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 99  
**E** 100
- 6** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměně A roven:

**A** 299  
**B** 200  
**C** 300  
**D** 101  
**E** kód obsahuje syntaktickou chybu  
**F** 100
- 7** Výsledkem výrazu (1 && 0) je:

**A** není definováno  
**B** 10  
**C** 0  
**D** 1
- 8** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

**A** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
```

Bude obsah proměně A roven:

**A** 300  
**B** 200  
**C** 100  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** kód obsahuje syntaktickou chybu
- 10** Hodnota výrazu (2+3)/5+5 je:

**A** 0  
**B** 6  
**C** 2  
**D** 1  
**E** 10
- 11** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

**A** 99x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 101x  
**D** 0x  
**E** 100x

**12**

Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A** 101x
- B** kód obsahuje syntaktickou chybu
- C** 99x
- D** 100x

**13**

Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** kód se nepřeloží (obsahuje chybu)
- B** "Hello World! , Hello World!!"
- C** "Hello World!"
- D** "!"
- E** "Hello World! , Hello World!"
- F** "Hello, Hello World!!"

**14**

Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A** 102
- B** kód obsahuje syntaktickou chybu
- C** 101
- D** 103



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		13

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 7 znacích a textový řetězec o délce 7 znaků
- B** Pole o 8 znacích a textový řetězec o délce 8 znaků
- C** Pole o 7 znacích a textový řetězec o délce 8 znaků
- D** Pole o 8 znacích a textový řetězec o délce 7 znaků
- E** kód se nepřeloží (obsahuje chybu)

**2** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
proběhne:
```

- A** 100x
- B** 101x
- C** kód obsahuje syntaktickou chybu
- D** 99x

**3** Výsledkem výrazu (1 && 1) je:

- A** 0
- B** 10
- C** 1
- D** není definováno

**4** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměně A roven:

- A** 101
- B** 102
- C** kód obsahuje syntaktickou chybu
- D** 100

**5** Následující zápis reprezentuje:

```
double x[10];
```

- A** Pole s 9 prvky s indexy 1..9 typu double
- B** Pole s 9 prvky s indexy 0..9 typu double
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole s 10 prvky s indexy 1..10 typu double
- E** Pole s 10 prvky s indexy 0..9 typu double

**6** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:

- A** 99x
- B** 0x
- C** 100x
- D** 101x
- E** kód se nepřeloží (obsahuje chybu)

**7** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** 6
- B** 3
- C** kód se nepřeloží (obsahuje chybu)
- D** 9
- E** 5

**8** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
```

Bude obsah proměně A roven:

- A** kód obsahuje syntaktickou chybu
- B** 299
- C** 300
- D** 101
- E** 200
- F** 100

**9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = pA + 200;
```

Bude obsah proměně A roven:

- A** 200
- B** kód obsahuje syntaktickou chybu
- C** obsah hodnoty A nelze bez spuštění jednoznačně určit
- D** 300
- E** 100

**10** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A** 40
- B** 0
- C** 4
- D** 60
- E** 30

- 11** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello, Hello World!!"  
**B** "!"  
**C** "Hello World! , Hello World!!"  
**D** "Hello World! , Hello World!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "Hello World!"

- 12** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++){
    { x -= 1; }
```
- A** -100  
**B** -99  
**C** 1  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 0

- 13** Datový typ int je:
- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**B** Typ bez znaménka pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel

- 14** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** 101x  
**B** 99x  
**C** kód obsahuje syntaktickou chybu  
**D** 100x

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 14           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 99x  
**D** 101x
- 2** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

**A** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**B** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**E** Pole o 8 znacích a textový řetězec o délce 7 znaků
- 3** Datový typ char je:

**A** Typ pro uchovávání velkých celých čísel  
**B** Typ pro uchovávání malých celých čísel nebo znaků  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- 4** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměnné A roven:

**A** 200  
**B** 100  
**C** 101  
**D** kód obsahuje syntaktickou chybu  
**E** 301  
**F** 300
- 5** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
```

Bude obsah proměnné A roven:

**A** 100  
**B** 200  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** 300  
**E** kód obsahuje syntaktickou chybu
- 6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++) {
    x -= 1;
}
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 0  
**C** 1  
**D** -99  
**E** -100
- 7** Následující zápis reprezentuje:

```
double x[10];
```

**A** Pole s 9 prvky s indexy 1..9 typu double  
**B** Pole s 10 prvky s indexy 1..10 typu double  
**C** Pole s 9 prvky s indexy 0..9 typu double  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** kód se nepřeloží (obsahuje chybu)
- 8** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "!"  
**B** "Hello World! , Hello World!"  
**C** "Hello, Hello World!!"  
**D** "Hello World!"  
**E** " , Hello World!!"  
**F** kód se nepřeloží (obsahuje chybu)
- 9** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:

**A** 0x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 100x  
**D** 101x  
**E** 99x
- 10** Výsledkem výrazu (1 || A), kde A je proměnná typu integer je:

**A** není definováno  
**B** 0 pokud je A rovno 0  
**C** 0  
**D** nenulová hodnota

**11** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A** 100
- B** 101
- C** kód obsahuje syntaktickou chybu
- D** 102

**12** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A** 4
- B** 60
- C** 30
- D** 40
- E** 0

**13** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměnná typu int proběhne:

- A** kód obsahuje syntaktickou chybu
- B** cyklus nemusí proběhnout ani jednou
- C** vždy 100x
- D** alespoň jednou

**14** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 5
- B** kód se nepřeloží (obsahuje chybu)
- C** 9
- D** 3
- E** 6

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 15           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  

```
int i = 99;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

  
**A** dvakrát  
**B** kód obsahuje syntaktickou chybu  
**C** jednou  
**D** ani jednou
- 
- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "Hello, Hello World!!"  
**B** "!"  
**C** "Hello World! , Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** " , Hello World!!"  
**F** "Hello World!"
- 
- 3** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}  
proběhne:
```

  
**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 99x  
**D** 101x
- 
- 4** Po provedení kódu:  

```
int A = 100;  
A = ++A + 200;
```

  
Bude obsah proměnné A roven:  
  
**A** 301  
**B** kód obsahuje syntaktickou chybu  
**C** 101  
**D** 100  
**E** 200  
**F** 300
- 
- 5** Hodnota výrazu  $(2+3)/5+5$  je:  
  
**A** 10  
**B** 0  
**C** 2  
**D** 1  
**E** 6

- 6** Datový typ char je:  
  
**A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**B** Typ pro uchovávání malých celých čísel nebo znaků  
**C** Typ pro uchovávání velkých celých čísel  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- 
- 7** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

  
**A** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 8** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
}  
proběhne:
```

  
**A** 0x  
**B** 100x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 99x  
**E** 101x
- 
- 9** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

  
**A** -1  
**B** 0  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 1  
**E** 101
- 
- 10** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}  
bude obsah proměnné A roven:
```

  
**A** 102  
**B** 103  
**C** 101  
**D** kód obsahuje syntaktickou chybu

**11** Výsledkem výrazu (1 && 0) je:

- A** není definováno
- B** 1
- C** 10
- D** 0

**12** Následující zápis reprezentuje:

```
double x[10];
```

- A** Pole s 9 prvky s indexy 1..9 typu double
- B** Pole s 10 prvky s indexy 0..9 typu double
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole s 10 prvky s indexy 1..10 typu double
- E** Pole s 9 prvky s indexy 0..9 typu double

**13** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;
```

Bude obsah proměnné A roven:

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** 300
- C** 200
- D** 100
- E** kód obsahuje syntaktickou chybu

**14** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** 3
- B** kód se nepřeloží (obsahuje chybu)
- C** 6
- D** 9
- E** 5

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 16           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Datový typ int je:

- A** Typ bez znaménka pro uchovávání celých čísel
- B** Typ se znaménkem pro uchovávání celých čísel
- C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**2** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello World! , Hello World!"
- B** "Hello World! , Hello World!"
- C** "Hello, Hello World!"
- D** kód se nepřeloží (obsahuje chybu)
- E** "!"
- F** "Hello World!"

**3** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** 0
- B** kód se nepřeloží (obsahuje chybu)
- C** 101
- D** 1
- E** -1

**4** Následující zápis reprezentuje:

```
double x[10];
```

- A** Pole s 10 prvky s indexy 1..10 typu double
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 9 prvky s indexy 0..9 typu double
- D** Pole s 10 prvky s indexy 0..9 typu double
- E** Pole s 9 prvky s indexy 1..9 typu double

**5** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměnná typu int proběhne:

- A** cyklus nemusí proběhnout ani jednou
- B** alespoň jednou
- C** vždy 100x
- D** kód obsahuje syntaktickou chybu

**6** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
Bude obsah proměnné A roven:
```

- A** kód obsahuje syntaktickou chybu
- B** 200
- C** obsah hodnoty A nelze bez spuštění jednoznačně určit
- D** 100
- E** 300

**7** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 6
- B** kód se nepřeloží (obsahuje chybu)
- C** 3
- D** 9
- E** 5

**8** Výsledkem výrazu (0 || A), kde A je proměnná typu integer je:

- A** 1
- B** není definováno
- C** 0
- D** 0 pokud je A rovno 0

**9** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole o 7 znacích a textový řetězec o délce 8 znaků
- D** Pole o 8 znacích a textový řetězec o délce 7 znaků
- E** Pole o 7 znacích a textový řetězec o délce 7 znaků

**10** Hodnota výrazu (2+3)/5+5 je:

- A** 1
- B** 2
- C** 6
- D** 0
- E** 10

- 11** Cyklus
- ```
int i = 0;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** 101x
  - C** 99x
  - D** 100x

- 12** Po provedení kódu:
- ```
int A = 100;
A = ++A + 200;
```
- Bude obsah proměné A roven:
- A** 200
  - B** 100
  - C** kód obsahuje syntaktickou chybu
  - D** 101
  - E** 301
  - F** 300

- 13** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```
- bude obsah proměné A roven:
- A** 97
  - B** 98
  - C** 101
  - D** 100
  - E** kód obsahuje syntaktickou chybu

- 14** Tělo cyklu
- ```
for (int i = 0; i < 100; i++) {
}
```
- proběhne:
- A** 101x
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 100x
  - D** 0x
  - E** 99x



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 17           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "!"
- C " , Hello World!!"
- D "Hello World! , Hello World!"
- E "Hello World!"
- F "Hello, Hello World!!"

**2** Po provedení kódu:  

```
int A = 100;  
A = A++ + 200;
```

Bude obsah proměnné A roven:

- A 101
- B 301
- C 300
- D 100
- E 200
- F kód obsahuje syntaktickou chybu

**3** Cyklus  

```
int i = 99;  
do {  
    i++;  
} while (i < 100);
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B dvakrát
- C ani jednou
- D jednou

**4** Následující zápis reprezentuje:  

```
int x[10];
```

- A Pole s 10 prvky s indexy 0..9 typu int
- B kód se nepřeloží (obsahuje chybu)
- C Pole s 9 prvky s indexy 0..9 typu int
- D Pole s 9 prvky s indexy 1..9 typu int
- E Pole s 10 prvky s indexy 1..10 typu int

**5** Tělo cyklu  

```
for (int i = 0; i <= 100; i++) {  
    }  
}
```

proběhne:

- A 100x
- B 0x
- C kód se nepřeloží (obsahuje chybu)
- D 99x
- E 101x

**6** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B 101x
- C 99x
- D 100x

**7** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }
```

- A kód se nepřeloží (obsahuje chybu)
- B -100
- C -99
- D 1
- E 0

**8** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 7 znacích a textový řetězec o délce 7 znaků
- C Pole o 7 znacích a textový řetězec o délce 8 znaků
- D Pole o 8 znacích a textový řetězec o délce 7 znaků
- E Pole o 8 znacích a textový řetězec o délce 8 znaků

**9** Výsledkem výrazu (1 || A), kde A je proměnná typu integer je:

- A 0
- B není definováno
- C 0 pokud je A rovno 0
- D nenulová hodnota

**10** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A 101
- B 100
- C 102
- D kód obsahuje syntaktickou chybu

**11** Datový typ int má vždy velikost:

- A 4 bajty
- B 2 bajty
- C 8 bajty
- D v závislosti na architektuře

**12** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 33
- B 44
- C 1
- D 66
- E 0

**13** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 3
- C 6
- D 9
- E 5

**14** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;  
Bude obsah proměnné A roven:
```

- A 200
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C kód obsahuje syntaktickou chybu
- D 300
- E 100

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 18           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 100;  
A = A-- + 200;  
Bude obsah proměné A roven:
```

- A 300
- B 299
- C kód obsahuje syntaktickou chybu
- D 200
- E 101
- F 100

**2** Cyklus

```
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:
```

- A 101x
- B kód obsahuje syntaktickou chybu
- C 100x
- D 99x

**3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 6
- C 3
- D 5
- E 9

**4** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 10 prvky s indexy 0..9 typu double
- B Pole s 9 prvky s indexy 1..9 typu double
- C Pole s 9 prvky s indexy 0..9 typu double
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 1..10 typu double

**5** Tělo cyklu

```
for (int i = 1; i < 100; i++) {  
}  
proběhne:
```

- A 101x
- B 100x
- C kód se nepřeloží (obsahuje chybu)
- D 99x
- E 0x

**6** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *A + 200;  
Bude obsah proměné A roven:
```

- A 300
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C 100
- D 200
- E kód obsahuje syntaktickou chybu

**7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World! , Hello World!!"
- B "Hello World! , Hello World!"
- C "Hello, Hello World!!"
- D "Hello World!"
- E kód se nepřeloží (obsahuje chybu)
- F "!"

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
for (int i = 0; i < 100; i++)  
    { x -= 1; }
```

- A -99
- B kód se nepřeloží (obsahuje chybu)
- C -100
- D 0
- E 1

**9** Výsledkem výrazu (1 || A), kde A je proměná typu integer je:

- A nenulová hodnota
- B není definováno
- C 0
- D 0 pokud je A rovno 0

**10** Datový typ int má vždy velikost:

- A 8 bajty
- B v závislosti na architektuře
- C 2 bajty
- D 4 bajty

- 11** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```
- bude obsah proměnné A roven:
- A** 101
  - B** 98
  - C** kód obsahuje syntaktickou chybu
  - D** 97
  - E** 100

- 12** Hodnota výrazu  $(2 - 20) / (5 + 5)$  je:
- A** 0
  - B** -1
  - C** 2
  - D** -1.8
  - E** 1

- 13** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - E** Pole o 6 znacích a textový řetězec o délce 5 znaků

- 14** Cyklus
- ```
do {
    i++;
} while (i < 100);
```
- kde i je proměnná typu int proběhne:
- A** cyklus nemusí proběhnout ani jednou
  - B** vždy 100x
  - C** kód obsahuje syntaktickou chybu
  - D** alespoň jednou

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		19

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = &pA + 200;
Bude obsah proměné A roven:
```

- A** kód obsahuje syntaktickou chybu
- B** obsah hodnoty A nelze bez spuštění jednoznačně určit
- C** 100
- D** 200
- E** 300

**2** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** 0
- B** -1
- C** kód se nepřeloží (obsahuje chybu)
- D** 101
- E** 1

**3** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello, Hello World!!"
- B** "Hello World! , Hello World!!"
- C** "Hello World!"
- D** "!"
- E** kód se nepřeloží (obsahuje chybu)
- F** "Hello World! , Hello World!"

**4** Tělo cyklu  
for (int i = 1; i < 100; i++) {  
}  
proběhne:

- A** kód se nepřeloží (obsahuje chybu)
- B** 100x
- C** 99x
- D** 0x
- E** 101x

**5** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A** -1
- B** 0
- C** 1
- D** 2
- E** -1.8

**6** Cyklus  
while (i < 100) {  
 i++;  
}  
kde i je proměná typu int proběhne:

- A** kód obsahuje syntaktickou chybu
- B** vždy 100x
- C** cyklus nemusí proběhnout ani jednou
- D** alespoň jednou

**7** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměná typu integer je:

- A** nenulová hodnota
- B** 0 pokud je A rovno 0
- C** není definováno
- D** 0

**8** Po provedení kódu:  
int A = 100;  
A = A-- + 200;  
Bude obsah proměné A roven:

- A** 200
- B** 100
- C** 300
- D** 101
- E** kód obsahuje syntaktickou chybu
- F** 299

**9** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 5
- C** 6
- D** 9
- E** 3

**10** Následující kód reprezentuje:  
char text[] = "Nazdar.";

- A** Pole o 8 znacích a textový řetězec o délce 7 znaků
- B** Pole o 7 znacích a textový řetězec o délce 8 znaků
- C** Pole o 7 znacích a textový řetězec o délce 7 znaků
- D** Pole o 8 znacích a textový řetězec o délce 8 znaků
- E** kód se nepřeloží (obsahuje chybu)

**11** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A** 98
- B** 102
- C** 101
- D** kód obsahuje syntaktickou chybu
- E** 100

**12** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměnná typu int proběhne:

- A** cyklus nemusí proběhnout ani jednou
- B** alespoň jednou
- C** kód obsahuje syntaktickou chybu
- D** vždy 100x

**13** Datový typ float je:

- A** Typ se znaménkem pro uchovávání celých čísel
- B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- C** Typ bez znaménka pro uchovávání celých čísel
- D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou

**14** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu int
- B** Pole s 10 prvky s indexy 1..10 typu int
- C** Pole s 9 prvky s indexy 1..9 typu int
- D** Pole s 9 prvky s indexy 0..9 typu int
- E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		20

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 2** Datový typ `int` je:
- A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou

- 3** Po vykonání následujícího kódu bude v `buf` uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!!");`
- A** "Hello World!"  
**B** "Hello, Hello World!!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "!"  
**E** "Hello World! , Hello World!!"  
**F** "Hello World! , Hello World!"

- 4** Cyklus  
`while (i < 100) {  
 i++;  
}`  
kde `i` je proměnná typu `int` proběhne:
- A** cyklus nemusí proběhnout ani jednou  
**B** vždy 100x  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu

- 5** Po provedení kódu:  
`int A = 100;  
switch (A) {  
 case 100: A++;  
 case 101: A++; break;  
 case 102: A++;  
}`  
bude obsah proměnné `A` roven:
- A** 102  
**B** 101  
**C** kód obsahuje syntaktickou chybu  
**D** 103

- 6** Cyklus  
`int i = 99;  
do {  
 i++;  
} while (i < 100);`  
proběhne:
- A** jednou  
**B** kód obsahuje syntaktickou chybu  
**C** ani jednou  
**D** dvakrát

- 7** Následující zápis reprezentuje:  
`int x[10];`
- A** Pole s 10 prvky s indexy 0..9 typu `int`  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 1..10 typu `int`  
**D** Pole s 9 prvky s indexy 1..9 typu `int`  
**E** Pole s 9 prvky s indexy 0..9 typu `int`

- 8** Po provedení kódu:  
`int A = 100;  
A = A++ + 200;`  
Bude obsah proměnné `A` roven:
- A** 100  
**B** 101  
**C** 300  
**D** 200  
**E** 301  
**F** kód obsahuje syntaktickou chybu

- 9** Hodnota proměnné `x` po provedení následující kódu bude:  
`int x = 0;  
if (1)  
 x = x + 1;  
else  
 x = x - 1;  
for (int i = 0; i < 100; i++) {x++};`
- A** -1  
**B** 1  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 101  
**E** 0

- 10** Výsledkem výrazu `(1 && 1)` je:
- A** není definováno  
**B** 0  
**C** 10  
**D** 1

- 11** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = **pA + 200;`  
Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** 300
  - C** 100
  - D** 200
  - E** kód obsahuje syntaktickou chybu

- 12** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 33
  - B** 1
  - C** 44
  - D** 0
  - E** 66

- 13** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {  
}  
}`  
proběhne:
- A** 100x
  - B** 101x
  - C** 99x
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 0x

- 14** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** 3
  - C** 5
  - D** 6
  - E** 9



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 21           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  
`int A = 100;  
A = ++A + 200;`  
Bude obsah proměnné A roven:

**A** 100  
**B** 300  
**C** 301  
**D** kód obsahuje syntaktickou chybu  
**E** 200  
**F** 101
- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");`

**A** "Hello World!"  
**B** "Hello, Hello World!"  
**C** " , Hello World!"  
**D** "!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "Hello World! , Hello World!"
- 3** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`

**A** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**E** Pole o 5 znacích a textový řetězec o délce 6 znaků
- 4** Datový typ int je:

**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**C** Typ se znaménkem pro uchovávání celých čísel  
**D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- 5** Cyklus  
`int i = 0;  
while (i <= 100) {  
 i++;  
}  
proběhne:`

**A** 99x  
**B** kód obsahuje syntaktickou chybu  
**C** 100x  
**D** 101x
- 6** Cyklus  
`int i = 0;  
do {  
 i++;  
} while (i < 100);  
proběhne:`

**A** 100x  
**B** kód obsahuje syntaktickou chybu  
**C** 99x  
**D** 101x
- 7** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

**A** 1  
**B** není definováno  
**C** 0 pokud je A rovno 0  
**D** 0
- 8** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
 y = y + x[i];`

**A** 5  
**B** 3  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 9  
**E** 6
- 9** Po provedení kódu:  
`int A = 100;  
switch (A) {  
 case 99: A++; break;  
 case 100: A++; break;  
 case 101: A++; break;  
}  
bude obsah proměnné A roven:`

**A** 101  
**B** 102  
**C** kód obsahuje syntaktickou chybu  
**D** 100
- 10** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }`

**A** kód se nepřeloží (obsahuje chybu)  
**B** -99  
**C** 1  
**D** -100  
**E** 0

- 
- 11** Následující zápis reprezentuje:  
`int x[10];`
- A** Pole s 9 prvky s indexy 1..9 typu int
  - B** Pole s 10 prvky s indexy 0..9 typu int
  - C** Pole s 9 prvky s indexy 0..9 typu int
  - D** Pole s 10 prvky s indexy 1..10 typu int
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 12** Tělo cyklu  
`for (int i = 0; i < 100; i++) {`  
`}`  
proběhne:
- A** 0x
  - B** 101x
  - C** 100x
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 99x
- 
- 13** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *A + 200;`  
Bude obsah proměnné A roven:
- A** 200
  - B** kód obsahuje syntaktickou chybu
  - C** 100
  - D** 300
  - E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 
- 14** Hodnota výrazu  $(2+3)/5+5$  je:
- A** 0
  - B** 10
  - C** 1
  - D** 2
  - E** 6
-

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 22           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 1  
**C** -99  
**D** 0  
**E** -100

- 2** Tělo cyklu
- ```
for (int i = 1; i <= 100; i++) {
}
```
- proběhne:

**A** 101x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 0x  
**D** 100x  
**E** 99x

- 3** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**E** Pole o 5 znacích a textový řetězec o délce 5 znaků

- 4** Výsledkem výrazu (1 || A), kde A je proměná typu integer je:

**A** nenulová hodnota  
**B** 0  
**C** není definováno  
**D** 0 pokud je A rovno 0

- 5** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** "Hello World!"  
**C** "!"  
**D** "Hello World! , Hello World!"  
**E** "Hello World! , Hello World!"  
**F** "Hello, Hello World!!"

- 6** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměně A roven:

**A** 300  
**B** 100  
**C** 301  
**D** 101  
**E** 200  
**F** kód obsahuje syntaktickou chybu

- 7** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 5  
**C** 3  
**D** 6  
**E** 9

- 8** Datový typ float je:

**A** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou  
**B** Typ se znaménkem pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání celých čísel  
**D** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou

- 9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměně A roven:

**A** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**B** kód obsahuje syntaktickou chybu  
**C** 300  
**D** 100  
**E** 200

- 10** Cyklus
- ```
while (i < 100) {
    i++;
}
```

kde i je proměná typu int proběhne:

**A** cyklus nemusí proběhnout ani jednou  
**B** kód obsahuje syntaktickou chybu  
**C** alespoň jednou  
**D** vždy 100x

**11** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 0..9 typu int
- B** Pole s 10 prvky s indexy 0..9 typu int
- C** Pole s 9 prvky s indexy 1..9 typu int
- D** Pole s 10 prvky s indexy 1..10 typu int
- E** kód se nepřeloží (obsahuje chybu)

**12** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A** 2
- B** 1
- C** 0
- D** -1
- E** -1.8

**13** Cyklus

```
int i = 100;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A** jednou
- B** kód obsahuje syntaktickou chybu
- C** dvakrát
- D** ani jednou

**14** Po provedení kódu:

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}  
bude obsah proměné A roven:
```

- A** 102
- B** 103
- C** 101
- D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 23           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 9 prvky s indexy 0..9 typu double
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole s 9 prvky s indexy 1..9 typu double
  - D** Pole s 10 prvky s indexy 0..9 typu double
  - E** Pole s 10 prvky s indexy 1..10 typu double

- 2** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = &pA + 200;`  
Bude obsah proměné A roven:
- A** 200
  - B** 100
  - C** 300
  - D** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - E** kód obsahuje syntaktickou chybu

- 3** Datový typ float je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - C** Typ se znaménkem pro uchovávání celých čísel
  - D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou

- 4** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 100: A++;`  
    `case 101: A++; break;`  
    `case 102: A++;`  
`}`  
bude obsah proměné A roven:
- A** 102
  - B** kód obsahuje syntaktickou chybu
  - C** 101
  - D** 103

- 5** Cyklus  
`do {`  
    `i++;`  
`} while (i < 100);`  
kde i je proměná typu int proběhne:
- A** vždy 100x
  - B** kód obsahuje syntaktickou chybu
  - C** alespoň jednou
  - D** cyklus nemusí proběhnout ani jednou

- 6** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** " , Hello World!!"
  - B** "!"
  - C** "Hello, Hello World!!"
  - D** "Hello World! , Hello World!"
  - E** "Hello World!"
  - F** kód se nepřeloží (obsahuje chybu)

- 7** Po provedení kódu:  
`int A = 100;`  
`A = --A + 200;`  
Bude obsah proměné A roven:
- A** 101
  - B** 200
  - C** 299
  - D** 100
  - E** 300
  - F** kód obsahuje syntaktickou chybu

- 8** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
    `i++;`  
`}`  
proběhne:
- A** 100x
  - B** 101x
  - C** kód obsahuje syntaktickou chybu
  - D** 99x

- 9** Hodnota výrazu  $(2+3)/5+5$  je:
- A** 10
  - B** 2
  - C** 6
  - D** 1
  - E** 0

- 10** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x += 1; }`
- A** 100
  - B** 0
  - C** 1
  - D** 99
  - E** kód se nepřeloží (obsahuje chybu)

- 11** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** 3  
**C** 5  
**D** 9  
**E** 6
- 
- 12** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků
- 
- 13** Tělo cyklu
- ```
for (int i = 0; i <= 100; i++) {  
    }  
}
```
- proběhne:
- A** 100x  
**B** 0x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 101x  
**E** 99x
- 
- 14** Výsledkem výrazu (1 && 0) je:
- A** není definováno  
**B** 10  
**C** 0  
**D** 1

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		24

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 6
- B kód se nepřeloží (obsahuje chybu)
- C 5
- D 9
- E 3

**2** Hodnota výrazu  $(2+3)/5+5$  je:

- A 2
- B 10
- C 1
- D 6
- E 0

**3** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello, Hello World!!"
- B kód se nepřeloží (obsahuje chybu)
- C "Hello World! , Hello World!"
- D "!"
- E " , Hello World!!"
- F "Hello World!"

**4** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }
```

- A kód se nepřeloží (obsahuje chybu)
- B 0
- C -100
- D -99
- E 1

**5** Po provedení kódu:

```
int A = 100;  
A = --A + 200;  
Bude obsah proměnné A roven:
```

- A kód obsahuje syntaktickou chybu
- B 299
- C 300
- D 200
- E 101
- F 100

**6** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 7 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 7 znacích a textový řetězec o délce 7 znaků
- D Pole o 8 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 8 znaků

**7** Cyklus  

```
int i = 0;  
do {  
    i++;  
} while (i < 100);
```

  
proběhne:

- A kód obsahuje syntaktickou chybu
- B 100x
- C 101x
- D 99x

**8** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

  
Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C 300
- D 200
- E 100

**9** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A 0 pokud je A rovno 0
- B 0
- C není definováno
- D nenulová hodnota

**10** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

  
proběhne:

- A 101x
- B 100x
- C kód obsahuje syntaktickou chybu
- D 99x

- 11** Datový typ char je:
- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - C** Typ pro uchovávání velkých celých čísel
  - D** Typ pro uchovávání malých celých čísel nebo znaků

- 12** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 10 prvků s indexy 1..10 typu double
  - B** Pole s 9 prvků s indexy 0..9 typu double
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole s 9 prvků s indexy 1..9 typu double
  - E** Pole s 10 prvků s indexy 0..9 typu double

- 13** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:
- A** 101x
  - B** 100x
  - C** 0x
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 99x

- 14** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
    `default: A--;`  
`}`  
bude obsah proměnné A roven:
- A** 98
  - B** 101
  - C** kód obsahuje syntaktickou chybu
  - D** 100
  - E** 97



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		25

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A 9
- B kód se nepřeloží (obsahuje chybu)
- C 6
- D 3
- E 5

**2** Datový typ int je:  
A Typ se znaménkem pro uchovávání celých čísel  
B Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
C Typ bez znaménka pro uchovávání celých čísel  
D Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**3** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = pA + 200;
```

Bude obsah proměnné A roven:  
A 200  
B 100  
C 300  
D kód obsahuje syntaktickou chybu  
E obsah hodnoty A nelze bez spuštění jednoznačně určit

**4** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

kde i je proměnná typu int proběhne:  
A kód obsahuje syntaktickou chybu  
B alespoň jednou  
C cyklus nemusí proběhnout ani jednou  
D vždy 100x

**5** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
A " , Hello World!!"  
B "Hello World! , Hello World!"  
C "!"  
D "Hello World!"  
E kód se nepřeloží (obsahuje chybu)  
F "Hello, Hello World!!"

**6** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
}
```

proběhne:  
A 0x  
B 99x  
C 100x  
D kód se nepřeloží (obsahuje chybu)  
E 101x

**7** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

  
A 0  
B 1  
C -1  
D kód se nepřeloží (obsahuje chybu)  
E 101

**8** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:  
A 0 pokud je A rovno 0  
B není definováno  
C nenulová hodnota  
D 0

**9** Následující zápis reprezentuje:  

```
double x[10];
```

  
A Pole s 9 prvky s indexy 1..9 typu double  
B Pole s 9 prvky s indexy 0..9 typu double  
C Pole s 10 prvky s indexy 0..9 typu double  
D kód se nepřeloží (obsahuje chybu)  
E Pole s 10 prvky s indexy 1..10 typu double

**10** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:  
A 0  
B 60  
C 30  
D 4  
E 40

- 11** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** kód obsahuje syntaktickou chybu
  - C** 100
  - D** 101
  - E** 200
  - F** 301

- 12** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 100: A++;`  
    `case 101: A++; break;`  
    `case 102: A++;`  
`}`  
bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 102
  - C** 101
  - D** 103

- 13** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - B** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - E** kód se nepřeloží (obsahuje chybu)

- 14** Cyklus  
`do {`  
    `i++;`  
`} while (i < 100);`  
kde i je proměnná typu int proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** alespoň jednou
  - C** cyklus nemusí proběhnout ani jednou
  - D** vždy 100x

# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		26

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Výsledkem výrazu (1 && 1) je:

- A není definováno
- B 0
- C 10
- D 1

**2** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A "!"
- B "Hello World!"
- C kód se nepřeloží (obsahuje chybu)
- D " , Hello World!!"
- E "Hello, Hello World!!"
- F "Hello World! , Hello World!"

**3** Datový typ float je:

- A Typ bez znaménka pro uchovávání celých čísel
- B Typ se znaménkem pro uchovávání celých čísel
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ se znaménkem pro uchovávání čísel s desetinou čárkou

**4** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
proběhne:
```

- A 101x
- B 100x
- C kód obsahuje syntaktickou chybu
- D 99x

**5** Po provedení kódu:

```
int A = 100;
A = --A + 200;
Bude obsah proměnné A roven:
```

- A 101
- B kód obsahuje syntaktickou chybu
- C 300
- D 299
- E 200
- F 100

**6** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A 6
- B kód se nepřeloží (obsahuje chybu)
- C 5
- D 9
- E 3

**7** Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 9 prvky s indexy 0..9 typu int
- B Pole s 10 prvky s indexy 0..9 typu int
- C Pole s 9 prvky s indexy 1..9 typu int
- D Pole s 10 prvky s indexy 1..10 typu int
- E kód se nepřeloží (obsahuje chybu)

**8** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {
}
proběhne:
```

- A 100x
- B kód se nepřeloží (obsahuje chybu)
- C 0x
- D 99x
- E 101x

**9** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

- A 0
- B 100
- C 1
- D kód se nepřeloží (obsahuje chybu)
- E 99

**10** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 5 znacích a textový řetězec o délce 6 znaků
- B Pole o 5 znacích a textový řetězec o délce 5 znaků
- C Pole o 6 znacích a textový řetězec o délce 5 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

- 11** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = **pA + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - C** 200
  - D** kód obsahuje syntaktickou chybu
  - E** 100

- 12** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:
- A** 4
  - B** 30
  - C** 60
  - D** 40
  - E** 0

- 13** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
`}`  
bude obsah proměnné A roven:
- A** 102
  - B** 100
  - C** kód obsahuje syntaktickou chybu
  - D** 101

- 14** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
    `i++;`  
`}`  
proběhne:
- A** 101x
  - B** 100x
  - C** kód obsahuje syntaktickou chybu
  - D** 99x

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		27

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus
- ```
int i = 0;
while (i < 100) {
    i++;
}
```
- proběhne:
- A** 99x
  - B** 100x
  - C** kód obsahuje syntaktickou chybu
  - D** 101x

- 2** Datový typ int má vždy velikost:
- A** v závislosti na architektuře
  - B** 8 bajty
  - C** 2 bajty
  - D** 4 bajty

- 3** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++) {
    x += 1;
}
```
- A** 100
  - B** 0
  - C** 1
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 99

- 4** Cyklus
- ```
int i = 0;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** 99x
  - C** 100x
  - D** 101x

- 5** Tělo cyklu
- ```
for (int i = 1; i < 100; i++) {
}
```
- proběhne:
- A** 101x
  - B** 100x
  - C** 0x
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 99x

- 6** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - E** kód se nepřeloží (obsahuje chybu)

- 7** Po provedení kódu:
- ```
int A = 100;
A = A-- + 200;
```
- Bude obsah proměnné A roven:
- A** 100
  - B** kód obsahuje syntaktickou chybu
  - C** 200
  - D** 299
  - E** 101
  - F** 300

- 8** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```
- bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 102
  - C** 103
  - D** 101

- 9** Výsledkem výrazu (1 && 0) je:
- A** 0
  - B** 1
  - C** 10
  - D** není definováno

- 10** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = *A + 200;
```
- Bude obsah proměnné A roven:
- A** 100
  - B** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - C** 300
  - D** kód obsahuje syntaktickou chybu
  - E** 200

**11** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 4
- B 40
- C 60
- D 30
- E 0

**12** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A 3
- B 6
- C kód se nepřeloží (obsahuje chybu)
- D 9
- E 5

**13** Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 10 prvky s indexy 1..10 typu int
- B Pole s 9 prvky s indexy 0..9 typu int
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 10 prvky s indexy 0..9 typu int
- E Pole s 9 prvky s indexy 1..9 typu int

**14** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World! , Hello World!"
- B "Hello World!"
- C " , Hello World!!"
- D kód se nepřeloží (obsahuje chybu)
- E "!"
- F "Hello, Hello World!!"

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		28

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 9 prvky s indexy 1..9 typu `double`
  - B** Pole s 10 prvky s indexy 0..9 typu `double`
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole s 9 prvky s indexy 0..9 typu `double`
  - E** Pole s 10 prvky s indexy 1..10 typu `double`

- 2** Po provedení kódu:  
`int A = 100;`  
`A = --A + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** 200
  - C** kód obsahuje syntaktickou chybu
  - D** 299
  - E** 100
  - F** 101

- 3** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:
- A** 101x
  - B** 100x
  - C** 0x
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 99x

- 4** Datový typ `float` je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - D** Typ se znaménkem pro uchovávání celých čísel

- 5** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole o 5 znacích a textový řetězec o délce 6 znaků

- 6** Cyklus  
`while (i < 100) {`  
`i++;`  
`}`  
kde `i` je proměnná typu `int` proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** vždy 100x
  - C** alespoň jednou
  - D** cyklus nemusí proběhnout ani jednou

- 7** Výsledkem výrazu `(1 && 0)` je:
- A** 10
  - B** 1
  - C** není definováno
  - D** 0

- 8** Hodnota výrazu `(2-20)/(5+5)` je:
- A** 1
  - B** 0
  - C** -1
  - D** -1.8
  - E** 2

- 9** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *pA + 200;`  
Bude obsah proměnné A roven:
- A** 100
  - B** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - C** 300
  - D** kód obsahuje syntaktickou chybu
  - E** 200

- 10** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
`case 100: A++;`  
`case 101: A++; break;`  
`case 102: A++;`  
`}`  
bude obsah proměnné A roven:
- A** 102
  - B** 101
  - C** kód obsahuje syntaktickou chybu
  - D** 103

- 11** Cyklus  
`int i = 0;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:
- A** 100x
  - B** 99x
  - C** kód obsahuje syntaktickou chybu
  - D** 101x

- 12** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```
- A** 99  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 1  
**D** 100  
**E** 0
- 
- 13** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** 9  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 6  
**D** 3  
**E** 5
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World! , Hello World!"  
**B** "Hello, Hello World!!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "Hello World!"  
**E** "Hello World! , Hello World!!"  
**F** "!"



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 29           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 1
- B 0
- C -1
- D -1.8
- E 2

**2** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

- A 0
- B 0 pokud je A rovno 0
- C není definováno
- D 1

**3** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
```

bude obsah proměnné A roven:

- A 98
- B 100
- C kód obsahuje syntaktickou chybu
- D 97
- E 101

**4** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 6 znacích a textový řetězec o délce 5 znaků
- B Pole o 5 znacích a textový řetězec o délce 6 znaků
- C Pole o 5 znacích a textový řetězec o délce 5 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

**5** Tělo cyklu

```
for (int i = 0; i < 100; i++) {
}
```

proběhne:

- A 101x
- B 0x
- C kód se nepřeloží (obsahuje chybu)
- D 99x
- E 100x

**6** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměnné A roven:

- A 200
- B 299
- C kód obsahuje syntaktickou chybu
- D 100
- E 101
- F 300

**7** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 5
- B kód se nepřeloží (obsahuje chybu)
- C 9
- D 6
- E 3

**8** Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

- A dvakrát
- B kód obsahuje syntaktickou chybu
- C jednou
- D ani jednou

**9** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A 101x
- B 99x
- C 100x
- D kód obsahuje syntaktickou chybu

- 10** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```
- A** -99  
**B** kód se nepřeloží (obsahuje chybu)  
**C** -100  
**D** 1  
**E** 0
- 
- 11** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World!"  
**B** "Hello World! , Hello World!"  
**C** "Hello, Hello World!!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello World! , Hello World!!"  
**F** "!"
- 
- 12** Následující zápis reprezentuje:
- ```
int x[10];
```
- A** Pole s 10 prvky s indexy 1..10 typu int  
**B** Pole s 10 prvky s indexy 0..9 typu int  
**C** Pole s 9 prvky s indexy 0..9 typu int  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 9 prvky s indexy 1..9 typu int
- 
- 13** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = *pA + 200;
```
- Bude obsah proměnné A roven:
- A** 100  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 200
- 
- 14** Datový typ float je:
- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**B** Typ bez znaménka pro uchovávání celých čísel  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 30           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující zápis reprezentuje:  
`double x[10];`

- A** Pole s 9 prvky s indexy 0..9 typu double
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 10 prvky s indexy 1..10 typu double
- D** Pole s 10 prvky s indexy 0..9 typu double
- E** Pole s 9 prvky s indexy 1..9 typu double

**2** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:

- A** kód se nepřeloží (obsahuje chybu)
- B** 100x
- C** 101x
- D** 99x
- E** 0x

**3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 9
- C** 6
- D** 5
- E** 3

**4** Cyklus  
`while (i < 100) {`  
 `i++;`  
`}`  
kde i je proměnná typu int proběhne:

- A** kód obsahuje syntaktickou chybu
- B** alespoň jednou
- C** cyklus nemusí proběhnout ani jednou
- D** vždy 100x

**5** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A** není definováno
- B** 10
- C** 0
- D** 1

**6** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 102
- C** 100
- D** 98
- E** 101

**7** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
for (int i = 0; i < 100; i++)  
    { x += 1; }
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 99
- C** 1
- D** 0
- E** 100

**8** Datový typ int je:

- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B** Typ se znaménkem pro uchovávání celých čísel
- C** Typ bez znaménka pro uchovávání celých čísel
- D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou

**9** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** "Hello World! , Hello World!"
- B** "Hello World! , Hello World!!"
- C** "Hello, Hello World!!"
- D** kód se nepřeloží (obsahuje chybu)
- E** "!"
- F** "Hello World!"

**10** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
- B** Pole o 7 znacích a textový řetězec o délce 8 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 7 znacích a textový řetězec o délce 7 znaků
- E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**11** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 0
- B 33
- C 1
- D 66
- E 44

**12** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = pA + 200;
```

Bude obsah proměnné A roven:

- A obsah hodnoty A nelze bez spuštění jednoznačně určit
- B 300
- C 100
- D 200
- E kód obsahuje syntaktickou chybu

**13** Po provedení kódu:

```
int A = 100;
```

```
A = A-- + 200;
```

Bude obsah proměnné A roven:

- A 100
- B 300
- C 299
- D 200
- E 101
- F kód obsahuje syntaktickou chybu

**14** Cyklus

```
int i = 0;
```

```
do {
```

```
    i++;
```

```
} while (i < 100);
```

proběhne:

- A 101x
- B 100x
- C 99x
- D kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 31           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  

```
int i = 99;  
do {  
    i++;  
} while (i < 100);
```

  
proběhne:  
**A** ani jednou  
**B** dvakrát  
**C** kód obsahuje syntaktickou chybu  
**D** jednou
- 2** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

  
**A** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**B** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 3** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
}
```

  
proběhne:  
**A** 100x  
**B** 0x  
**C** 101x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 99x
- 4** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = pA + 200;
```

  
Bude obsah proměnné A roven:  
**A** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**B** 300  
**C** 100  
**D** 200  
**E** kód obsahuje syntaktickou chybu
- 5** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}
```

  
proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 101x  
**D** 99x
- 6** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++){  
    { x -= 1; }
```

  
**A** 1  
**B** -99  
**C** -100  
**D** 0  
**E** kód se nepřeloží (obsahuje chybu)
- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "Hello World!"  
**B** " , Hello World!!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "Hello World! , Hello World!"  
**E** "Hello, Hello World!!"  
**F** "!"
- 8** Po provedení kódu:  

```
int A = 100;  
A = --A + 200;
```

  
Bude obsah proměnné A roven:  
**A** 200  
**B** 300  
**C** 101  
**D** 100  
**E** kód obsahuje syntaktickou chybu  
**F** 299
- 9** Následující zápis reprezentuje:  

```
int x[10];
```

  
**A** Pole s 9 prvky s indexy 0..9 typu int  
**B** Pole s 9 prvky s indexy 1..9 typu int  
**C** Pole s 10 prvky s indexy 1..10 typu int  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 0..9 typu int
- 10** Výsledkem výrazu (1 && 0) je:  
**A** 10  
**B** není definováno  
**C** 0  
**D** 1

- 11** Datový typ float je:
- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - C** Typ se znaménkem pro uchovávání celých čísel
  - D** Typ bez znaménka pro uchovávání celých čísel

- 12** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```
- bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 97
  - C** 101
  - D** 98
  - E** 100

- 13** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 33
  - B** 0
  - C** 1
  - D** 66
  - E** 44

- 14** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** 6
  - C** 5
  - D** 3
  - E** 9

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 32           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = pA + 200;`  
Bude obsah proměnné A roven:
- A** 200
  - B** 300
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** kód obsahuje syntaktickou chybu
  - E** 100

- 2** Po provedení kódu:  
`int A = 98;  
switch (A) {  
 case 99: A++; break;  
 case 100: A++; break;  
 case 101: A++; break;  
 default: A--;  
}`  
bude obsah proměnné A roven:
- A** 97
  - B** 100
  - C** 101
  - D** 98
  - E** kód obsahuje syntaktickou chybu

- 3** Výsledkem výrazu `(1 || A)`, kde A je proměnná typu integer je:
- A** není definováno
  - B** 0
  - C** 0 pokud je A rovno 0
  - D** nenulová hodnota

- 4** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
 y = y + x[i];`
- A** 5
  - B** 9
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 6
  - E** 3

- 5** Cyklus  
`int i = 0;  
while (i <= 100) {  
 i++;  
}`  
proběhne:
- A** 100x
  - B** 101x
  - C** 99x
  - D** kód obsahuje syntaktickou chybu

- 6** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++)  
 { x += 1; }`
- A** 99
  - B** 0
  - C** 100
  - D** 1
  - E** kód se nepřeloží (obsahuje chybu)

- 7** Tělo cyklu  
`for (int i = 0; i < 100; i++) {  
}`  
proběhne:
- A** 0x
  - B** 100x
  - C** 99x
  - D** 101x
  - E** kód se nepřeloží (obsahuje chybu)

- 8** Cyklus  
`int i = 0;  
do {  
 i++;  
} while (i < 100);`  
proběhne:
- A** 99x
  - B** 100x
  - C** kód obsahuje syntaktickou chybu
  - D** 101x

- 9** Po provedení kódu:  
`int A = 100;  
A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 301
  - B** 200
  - C** 101
  - D** 300
  - E** 100
  - F** kód obsahuje syntaktickou chybu

- 10** Datový typ float je:
- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - C** Typ bez znaménka pro uchovávání celých čísel
  - D** Typ se znaménkem pro uchovávání celých čísel

- 11** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World!"
  - B** "Hello World! , Hello World!!"
  - C** kód se nepřeloží (obsahuje chybu)
  - D** "!"
  - E** "Hello World! , Hello World!"
  - F** "Hello, Hello World!!"

- 12** Hodnota výrazu  $(2+3)/5+5$  je:
- A** 0
  - B** 2
  - C** 1
  - D** 6
  - E** 10

- 13** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - C** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - D** Pole o 8 znacích a textový řetězec o délce 7 znaků
  - E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 14** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 9 prvky s indexy 0..9 typu double
  - B** Pole s 9 prvky s indexy 1..9 typu double
  - C** Pole s 10 prvky s indexy 0..9 typu double
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole s 10 prvky s indexy 1..10 typu double



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		33

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A " , Hello World!!"
- B "Hello, Hello World!!"
- C kód se nepřeloží (obsahuje chybu)
- D "Hello World! , Hello World!"
- E "Hello World!"
- F "!"

**2** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = pA + 200;
```

Bude obsah proměnné A roven:

- A 100
- B 200
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 300
- E kód obsahuje syntaktickou chybu

**3** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x += 1; }
```

- A kód se nepřeloží (obsahuje chybu)
- B 0
- C 100
- D 1
- E 99

**4** Následující zápis reprezentuje:  

```
double x[10];
```

- A Pole s 10 prvky s indexy 1..10 typu double
- B Pole s 10 prvky s indexy 0..9 typu double
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 9 prvky s indexy 1..9 typu double
- E Pole s 9 prvky s indexy 0..9 typu double

**5** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 8 znaků
- B Pole o 7 znacích a textový řetězec o délce 7 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 8 znacích a textový řetězec o délce 7 znaků
- E Pole o 7 znacích a textový řetězec o délce 8 znaků

**6** Výsledkem výrazu (1 && 1) je:

- A 1
- B 10
- C není definováno
- D 0

**7** Po provedení kódu:  

```
int A = 100;  
A = A++ + 200;
```

Bude obsah proměnné A roven:

- A 100
- B 200
- C 300
- D 101
- E kód obsahuje syntaktickou chybu
- F 301

**8** Datový typ float je:

- A Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ bez znaménka pro uchovávání celých čísel
- D Typ se znaménkem pro uchovávání celých čísel

**9** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B 99x
- C 101x
- D 100x

**10** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 60
- B 40
- C 30
- D 4
- E 0

**11** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
}
```

proběhne:

- A 100x
- B 0x
- C 101x
- D kód se nepřeloží (obsahuje chybu)
- E 99x

- 12** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 3  
**B** 9  
**C** 5  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 6

- 13** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```
- bude obsah proměnné A roven:
- A** 103  
**B** 101  
**C** kód obsahuje syntaktickou chybu  
**D** 102

- 14** Cyklus
- ```
int i = 99;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** jednou  
**C** dvakrát  
**D** ani jednou

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 34           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`  
**A** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 5 znacích a textový řetězec o délce 5 znaků
- 2** Datový typ `int` je:  
**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**C** Typ se znaménkem pro uchovávání celých čísel  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- 3** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměnné `A` roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 300  
**D** obsah hodnoty `A` nelze bez spuštění jednoznačně určit  
**E** 200
- 4** Po provedení kódu:  
`int A = 98;  
switch (A) {  
 case 99: A++; break;  
 case 100: A++; break;  
 case 101: A++; break;  
}`  
bude obsah proměnné `A` roven:  
**A** 100  
**B** 101  
**C** 102  
**D** kód obsahuje syntaktickou chybu  
**E** 98
- 5** Cyklus  
`while (i < 100) {  
 i++;  
}`  
kde `i` je proměnná typu `int` proběhne:  
**A** vždy 100x  
**B** alespoň jednou  
**C** cyklus nemusí proběhnout ani jednou  
**D** kód obsahuje syntaktickou chybu
- 6** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {  
}`  
proběhne:  
**A** 0x  
**B** 100x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 99x  
**E** 101x
- 7** Výsledkem výrazu  $(1 \parallel A)$ , kde `A` je proměnná typu `integer` je:  
**A** není definováno  
**B** 0  
**C** nenulová hodnota  
**D** 0 pokud je `A` rovno 0
- 8** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 2  
**B** 6  
**C** 1  
**D** 10  
**E** 0
- 9** Následující zápis reprezentuje:  
`double x[10];`  
**A** Pole s 10 prvky s indexy 0..9 typu `double`  
**B** Pole s 10 prvky s indexy 1..10 typu `double`  
**C** Pole s 9 prvky s indexy 1..9 typu `double`  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 9 prvky s indexy 0..9 typu `double`
- 10** Cyklus  
`do {  
 i++;  
} while (i < 100);`  
kde `i` je proměnná typu `int` proběhne:  
**A** alespoň jednou  
**B** kód obsahuje syntaktickou chybu  
**C** vždy 100x  
**D** cyklus nemusí proběhnout ani jednou
- 11** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
 y = y + x[i];`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 6  
**C** 9  
**D** 3  
**E** 5

**12** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!!");
```

- A** "Hello World! , Hello World!"
- B** kód se nepřeloží (obsahuje chybu)
- C** "!"
- D** "Hello, Hello World!!"
- E** "Hello World!"
- F** "Hello World! , Hello World!!"

**13** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
Bude obsah proměnné A roven:
```

- A** 100
- B** 301
- C** 300
- D** 101
- E** 200
- F** kód obsahuje syntaktickou chybu

**14** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++){
    { x += 1; }
```

- A** 1
- B** 100
- C** kód se nepřeloží (obsahuje chybu)
- D** 99
- E** 0

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 35           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```

- A -99
- B 0
- C kód se nepřeloží (obsahuje chybu)
- D -100
- E 1

**2** Po provedení kódu:  

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A 101
- B 102
- C 103
- D kód obsahuje syntaktickou chybu

**3** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 9
- B 5
- C kód se nepřeloží (obsahuje chybu)
- D 3
- E 6

**4** Datový typ int je:

- A Typ bez znaménka pro uchovávání celých čísel
- B Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- C Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D Typ se znaménkem pro uchovávání celých čísel

**5** Cyklus  

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B 101x
- C 100x
- D 99x

**6** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 7 znacích a textový řetězec o délce 7 znaků
- C Pole o 8 znacích a textový řetězec o délce 8 znaků
- D Pole o 8 znacích a textový řetězec o délce 7 znaků
- E Pole o 7 znacích a textový řetězec o délce 8 znaků

**7** Hodnota výrazu (2+3)/5+5 je:

- A 10
- B 1
- C 0
- D 2
- E 6

**8** Cyklus  

```
int i = 100;
do {
    i++;
} while (i < 100);
```

proběhne:

- A jednou
- B dvakrát
- C ani jednou
- D kód obsahuje syntaktickou chybu

**9** Výsledkem výrazu (1 && 0) je:

- A 1
- B 0
- C 10
- D není definováno

**10** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A obsah hodnoty A nelze bez spuštění jednoznačně určit
- B 200
- C 100
- D kód obsahuje syntaktickou chybu
- E 300

**11** Následující zápis reprezentuje:  

```
int x[10];
```

- A Pole s 10 prvky s indexy 1..10 typu int
- B Pole s 9 prvky s indexy 1..9 typu int
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 9 prvky s indexy 0..9 typu int
- E Pole s 10 prvky s indexy 0..9 typu int

- 12** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:
- A** 100x
  - B** 0x
  - C** 101x
  - D** 99x
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Po vykonání následujícího kódu bude v `buf` uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** " , Hello World!!"
  - B** kód se nepřeloží (obsahuje chybu)
  - C** "Hello World!"
  - D** "Hello World! , Hello World!"
  - E** "Hello, Hello World!!"
  - F** "!"
- 
- 14** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné `A` roven:
- A** 301
  - B** 101
  - C** 300
  - D** 200
  - E** 100
  - F** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 36           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  

```
int i = 0;  
do {  
    i++;  
} while (i < 100);
```

  
proběhne:  
**A** 101x  
**B** 100x  
**C** 99x  
**D** kód obsahuje syntaktickou chybu

- 2** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

  
kde i je proměná typu int proběhne:  
**A** vždy 100x  
**B** kód obsahuje syntaktickou chybu  
**C** cyklus nemusí proběhnout ani jednou  
**D** alespoň jednou

- 3** Datový typ float je:  
**A** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou  
**B** Typ se znaménkem pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání celých čísel  
**D** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou

- 4** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
}
```

  
proběhne:  
**A** 0x  
**B** 101x  
**C** 99x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 100x

- 5** Po provedení kódu:  

```
int A = 100;  
A = --A + 200;
```

  
Bude obsah proměné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 300  
**C** 299  
**D** 100  
**E** 101  
**F** 200

- 6** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

  
**A** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 7** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;
```

  
Bude obsah proměné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 200  
**D** 100  
**E** 300

- 8** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }
```

  
**A** 1  
**B** 0  
**C** -100  
**D** -99  
**E** kód se nepřeloží (obsahuje chybu)

- 9** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 6  
**C** 5  
**D** 9  
**E** 3

- 10** Hodnota výrazu  $(2/3) \cdot (10+11+12) \cdot 2$  je:  
**A** 0  
**B** 66  
**C** 1  
**D** 33  
**E** 44

- 11** Následující zápis reprezentuje:  
`int x[10];`
- A** Pole s 9 prvky s indexy 1..9 typu int
  - B** Pole s 10 prvky s indexy 0..9 typu int
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole s 9 prvky s indexy 0..9 typu int
  - E** Pole s 10 prvky s indexy 1..10 typu int
- 
- 12** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** není definováno
  - B** nenulová hodnota
  - C** 0 pokud je A rovno 0
  - D** 0
- 
- 13** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello, Hello World!!"
  - B** "Hello World! , Hello World!!"
  - C** "Hello World! , Hello World!"
  - D** kód se nepřeloží (obsahuje chybu)
  - E** "!"
  - F** "Hello World!"
- 
- 14** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 100
  - C** 102
  - D** 101
  - E** 98



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 37           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměné A roven:
- A** 100  
**B** kód obsahuje syntaktickou chybu  
**C** 102  
**D** 101
- 2** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** 3  
**C** 6  
**D** 5  
**E** 9
- 3** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**B** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 5 znacích a textový řetězec o délce 6 znaků
- 4** Tělo cyklu  

```
for (int i = 0; i <= 100; i++) {  
}
```

proběhne:
- A** 101x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 99x  
**D** 0x  
**E** 100x
- 5** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = pA + 200;
```

Bude obsah proměné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**B** 100  
**C** 200  
**D** kód obsahuje syntaktickou chybu  
**E** 300

- 6** Výsledkem výrazu (1 && 1) je:
- A** 0  
**B** není definováno  
**C** 1  
**D** 10
- 7** Hodnota výrazu (2+3)/5+5 je:
- A** 0  
**B** 1  
**C** 10  
**D** 2  
**E** 6
- 8** Následující zápis reprezentuje:  

```
int x[10];
```
- A** Pole s 10 prvky s indexy 1..10 typu int  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 9 prvky s indexy 1..9 typu int  
**D** Pole s 10 prvky s indexy 0..9 typu int  
**E** Pole s 9 prvky s indexy 0..9 typu int
- 9** Cyklus  

```
do {  
    i++;  
} while (i < 100);
```

kde i je proměná typu int proběhne:
- A** cyklus nemusí proběhnout ani jednou  
**B** vždy 100x  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu
- 10** Po provedení kódu:  

```
int A = 100;  
A = ++A + 200;
```

Bude obsah proměné A roven:
- A** 301  
**B** 101  
**C** 300  
**D** kód obsahuje syntaktickou chybu  
**E** 100  
**F** 200
- 11** Datový typ float je:
- A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání celých čísel  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

- 12** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
 `i++;`  
`}`  
proběhne:
- A** 99x  
**B** 101x  
**C** 100x  
**D** kód obsahuje syntaktickou chybu
- 
- 13** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x -= 1; }`
- A** -99  
**B** 0  
**C** kód se nepřeloží (obsahuje chybu)  
**D** -100  
**E** 1
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** " , Hello World!!"  
**B** "Hello, Hello World!!"  
**C** "Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello World! , Hello World!"  
**F** "!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 38           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** alespoň jednou  
**B** cyklus nemusí proběhnout ani jednou  
**C** vždy 100x  
**D** kód obsahuje syntaktickou chybu

- 2** Po provedení kódu:  
int A = 100;  
int\* pA = &A;  
A = &pA + 200;  
Bude obsah proměné A roven:  
**A** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**B** kód obsahuje syntaktickou chybu  
**C** 100  
**D** 300  
**E** 200

- 3** Cyklus  
while (i < 100) {  
    i++;  
}  
kde i je proměná typu int proběhne:  
**A** cyklus nemusí proběhnout ani jednou  
**B** kód obsahuje syntaktickou chybu  
**C** alespoň jednou  
**D** vždy 100x

- 4** Tělo cyklu  
for (int i = 0; i < 100; i++) {  
}  
proběhne:  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 101x  
**C** 100x  
**D** 99x  
**E** 0x

- 5** Následující zápis reprezentuje:  
int x[10];  
**A** Pole s 9 prvky s indexy 0..9 typu int  
**B** Pole s 10 prvky s indexy 1..10 typu int  
**C** Pole s 9 prvky s indexy 1..9 typu int  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 0..9 typu int

- 6** Po provedení kódu:  
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 101  
**C** 102  
**D** 100

- 7** Výsledkem výrazu (1 && 1) je:  
**A** 10  
**B** není definováno  
**C** 1  
**D** 0

- 8** Po vykonání následujícího kódu bude v buf uložen řetězec:  
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");  
**A** "Hello, Hello World!!"  
**B** "!"  
**C** "Hello World!"  
**D** "Hello World! , Hello World!!"  
**E** "Hello World! , Hello World!"  
**F** kód se nepřeloží (obsahuje chybu)

- 9** Hodnota proměnné x po provedení následující kódu bude:  
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }  
**A** -100  
**B** -99  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 0  
**E** 1

- 10** Hodnota proměnné po provedení následujícího kódu bude:  
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 9  
**C** 3  
**D** 6  
**E** 5

- 11** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - E** Pole o 6 znacích a textový řetězec o délce 5 znaků
- 
- 12** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** 1
  - B** 0
  - C** -1
  - D** 2
  - E** -1.8
- 
- 13** Datový typ float je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - C** Typ se znaménkem pro uchovávání celých čísel
  - D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- 
- 14** Po provedení kódu:  
`int A = 100;`  
`A = A-- + 200;`  
Bude obsah proměnné A roven:
- A** 100
  - B** 299
  - C** 101
  - D** 200
  - E** 300
  - F** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 39           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  
`for (int i = 1; i < 100; i++) {  
}  
proběhne:`

**A** 0x  
**B** 100x  
**C** 99x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101x
- 2** Datový typ `int` je:

**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel
- 3** Výsledkem výrazu `(1 && 1)` je:

**A** 0  
**B** 1  
**C** 10  
**D** není definováno
- 4** Cyklus  
`int i = 100;  
do {  
    i++;  
} while (i < 100);  
proběhne:`

**A** kód obsahuje syntaktickou chybu  
**B** ani jednou  
**C** jednou  
**D** dvakrát
- 5** Následující zápis reprezentuje:  
`double x[10];`

**A** Pole s 10 prvky s indexy 0..9 typu `double`  
**B** Pole s 10 prvky s indexy 1..10 typu `double`  
**C** Pole s 9 prvky s indexy 1..9 typu `double`  
**D** Pole s 9 prvky s indexy 0..9 typu `double`  
**E** kód se nepřeloží (obsahuje chybu)
- 6** Hodnota výrazu `(2-20)/(5+5)` je:

**A** -1.8  
**B** 0  
**C** -1  
**D** 2  
**E** 1
- 7** Hodnota proměnné `x` po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++);  
{ x -= 1; }`

**A** 0  
**B** kód se nepřeloží (obsahuje chybu)  
**C** -99  
**D** -100  
**E** 1
- 8** Po provedení kódu:  
`int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměnné A roven:`

**A** 98  
**B** kód obsahuje syntaktickou chybu  
**C** 100  
**D** 102  
**E** 101
- 9** Cyklus  
`int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:`

**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 99x  
**D** 101x
- 10** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];`

**A** kód se nepřeloží (obsahuje chybu)  
**B** 5  
**C** 3  
**D** 9  
**E** 6

- 11** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = &pA + 200;`  
Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** 200
  - C** 100
  - D** kód obsahuje syntaktickou chybu
  - E** 300
- 
- 13** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 301
  - B** kód obsahuje syntaktickou chybu
  - C** 200
  - D** 101
  - E** 300
  - F** 100
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcat(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** "!"
  - B** "Hello World!"
  - C** "Hello World! , Hello World!"
  - D** "Hello World! , Hello World!!"
  - E** kód se nepřeloží (obsahuje chybu)
  - F** "Hello, Hello World!!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 40           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** 0 pokud je A rovno 0
  - B** nenulová hodnota
  - C** 0
  - D** není definováno

- 2** Následující zápis reprezentuje: `double x[10];`
- A** Pole s 10 prvky s indexy 1..10 typu double
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole s 9 prvky s indexy 0..9 typu double
  - D** Pole s 10 prvky s indexy 0..9 typu double
  - E** Pole s 9 prvky s indexy 1..9 typu double

- 3** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 44
  - B** 1
  - C** 0
  - D** 66
  - E** 33

- 4** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = *pA + 200;
```
- Bude obsah proměnné A roven:
- A** 300
  - B** 100
  - C** 200
  - D** kód obsahuje syntaktickou chybu
  - E** obsah hodnoty A nelze bez spuštění jednoznačně určit

- 5** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "!"
  - B** "Hello, Hello World!"
  - C** " , Hello World!"
  - D** "Hello World!"
  - E** "Hello World! , Hello World!"
  - F** kód se nepřeloží (obsahuje chybu)

- 6** Cyklus
- ```
do {
    i++;
} while (i < 100);
```
- kde i je proměnná typu int proběhne:
- A** cyklus nemusí proběhnout ani jednou
  - B** kód obsahuje syntaktickou chybu
  - C** vždy 100x
  - D** alespoň jednou

- 7** Datový typ int je:
- A** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
  - B** Typ se znaménkem pro uchovávání celých čísel
  - C** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
  - D** Typ bez znaménka pro uchovávání celých čísel

- 8** Cyklus
- ```
int i = 0;
while (i < 100) {
    i++;
}
```
- proběhne:
- A** 100x
  - B** 99x
  - C** 101x
  - D** kód obsahuje syntaktickou chybu

- 9** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```
- A** 0
  - B** -100
  - C** 1
  - D** kód se nepřeloží (obsahuje chybu)
  - E** -99

- 10** Po provedení kódu:
- ```
int A = 100;
A = --A + 200;
```
- Bude obsah proměnné A roven:
- A** 101
  - B** 299
  - C** 300
  - D** 200
  - E** 100
  - F** kód obsahuje syntaktickou chybu

**11** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A** 102
- B** 101
- C** kód obsahuje syntaktickou chybu
- D** 100

**12** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 6 znacích a textový řetězec o délce 5 znaků
- B** Pole o 6 znacích a textový řetězec o délce 6 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 5 znacích a textový řetězec o délce 6 znaků
- E** Pole o 5 znacích a textový řetězec o délce 5 znaků

**13** Tělo cyklu

```
for (int i = 0; i < 100; i++) {
}
```

proběhne:

- A** 99x
- B** 0x
- C** kód se nepřeloží (obsahuje chybu)
- D** 100x
- E** 101x

**14** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** 6
- B** 3
- C** 9
- D** kód se nepřeloží (obsahuje chybu)
- E** 5



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 41           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  

```
int i = 0;
do {
    i++;
} while (i < 100);
```

proběhne:  
**A** 99x  
**B** 100x  
**C** kód obsahuje syntaktickou chybu  
**D** 101x
- 2** Cyklus  

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:  
**A** 99x  
**B** kód obsahuje syntaktickou chybu  
**C** 101x  
**D** 100x
- 3** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 6  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 5  
**D** 3  
**E** 9
- 4** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!!");
```

**A** "Hello, Hello World!!"  
**B** "Hello World! , Hello World!"  
**C** "Hello World!"  
**D** "Hello World! , Hello World!!"  
**E** "!"  
**F** kód se nepřeloží (obsahuje chybu)
- 5** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 66  
**B** 1  
**C** 44  
**D** 0  
**E** 33

- 6** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**E** Pole o 5 znacích a textový řetězec o délce 5 znaků
- 7** Následující zápis reprezentuje:  

```
int x[10];
```

**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** Pole s 10 prvky s indexy 0..9 typu int  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 9 prvky s indexy 0..9 typu int  
**E** Pole s 9 prvky s indexy 1..9 typu int
- 8** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {
}
```

proběhne:  
**A** 100x  
**B** 101x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 99x  
**E** 0x
- 9** Datový typ int má vždy velikost:  
**A** v závislosti na architektuře  
**B** 2 bajty  
**C** 8 bajty  
**D** 4 bajty
- 10** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

**A** 1  
**B** -1  
**C** 101  
**D** 0  
**E** kód se nepřeloží (obsahuje chybu)
- 11** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:  
**A** 0  
**B** 1  
**C** není definováno  
**D** 0 pokud je A rovno 0

- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = &pA + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 200
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 100
  - E** 300

- 13** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 101
  - C** 100
  - D** 200
  - E** 301
  - F** 300

- 14** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 100: A++;`  
    `case 101: A++; break;`  
    `case 102: A++;`  
`}`  
bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 102
  - C** 103
  - D** 101

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 42           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`  
**A** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků
- 2** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:  
**A** 10  
**B** 1  
**C** 0  
**D** není definováno
- 3** Cyklus  
`int i = 0;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:  
**A** 101x  
**B** 100x  
**C** 99x  
**D** kód obsahuje syntaktickou chybu
- 4** Cyklus  
`while (i < 100) {`  
`i++;`  
`}`  
kde *i* je proměnná typu `int` proběhne:  
**A** alespoň jednou  
**B** cyklus nemusí proběhnout ani jednou  
**C** vždy 100x  
**D** kód obsahuje syntaktickou chybu
- 5** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {`  
`}`  
proběhne:  
**A** 100x  
**B** 0x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 99x  
**E** 101x
- 6** Následující zápis reprezentuje:  
`double x[10];`  
**A** Pole s 9 prvky s indexy 0..9 typu `double`  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 0..9 typu `double`  
**D** Pole s 10 prvky s indexy 1..10 typu `double`  
**E** Pole s 9 prvky s indexy 1..9 typu `double`
- 7** Hodnota proměnné *x* po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x += 1; }`  
**A** 99  
**B** 0  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 1  
**E** 100
- 8** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *A + 200;`  
Bude obsah proměnné *A* roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 300  
**C** 200  
**D** 100  
**E** obsah hodnoty *A* nelze bez spuštění jednoznačně určit
- 9** Datový typ `int` má vždy velikost:  
**A** 2 bajty  
**B** 4 bajty  
**C** v závislosti na architektuře  
**D** 8 bajty
- 10** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:  
**A** 30  
**B** 60  
**C** 40  
**D** 0  
**E** 4
- 11** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
`case 100: A++;`  
`case 101: A++; break;`  
`case 102: A++;`  
`}`  
bude obsah proměnné *A* roven:  
**A** 102  
**B** kód obsahuje syntaktickou chybu  
**C** 101  
**D** 103

- 12** Po provedení kódu:  
`int A = 100;`  
`A = A-- + 200;`  
Bude obsah proměnné A roven:  
**A** 101  
**B** 100  
**C** 300  
**D** kód obsahuje syntaktickou chybu  
**E** 200  
**F** 299
- 
- 13** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i <= 3; i++)`  
`y = y + x[i];`  
**A** 3  
**B** 9  
**C** 6  
**D** 5  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`  
**A** "Hello World!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** " , Hello World!!"  
**D** "Hello, Hello World!!"  
**E** "Hello World! , Hello World!"  
**F** "!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 43           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 6 znacích a textový řetězec o délce 5 znaků
- B** Pole o 6 znacích a textový řetězec o délce 6 znaků
- C** Pole o 5 znacích a textový řetězec o délce 5 znaků
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole o 5 znacích a textový řetězec o délce 6 znaků

**2** Po provedení kódu:

```
int A = 100;
```

```
A = A++ + 200;
```

Bude obsah proměnné A roven:

- A** 101
- B** 300
- C** kód obsahuje syntaktickou chybu
- D** 301
- E** 100
- F** 200

**3** Po provedení kódu:

```
int A = 100;
```

```
switch (A) {
```

```
    case 99: A++; break;
```

```
    case 100: A++; break;
```

```
    case 101: A++; break;
```

```
}
```

bude obsah proměnné A roven:

- A** 101
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 102

**4** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = pA + 200;
```

Bude obsah proměnné A roven:

- A** 300
- B** kód obsahuje syntaktickou chybu
- C** 200
- D** obsah hodnoty A nelze bez spuštění jednoznačně určit
- E** 100

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
```

```
int y = 0;
```

```
for (int i = 1; i < 3; i++)
```

```
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 6
- C** 9
- D** 5
- E** 3

**6** Cyklus

```
while (i < 100) {
```

```
    i++;
```

```
}
```

kde i je proměnná typu int proběhne:

- A** cyklus nemusí proběhnout ani jednou
- B** kód obsahuje syntaktickou chybu
- C** vždy 100x
- D** alespoň jednou

**7** Tělo cyklu

```
for (int i = 0; i < 100; i++) {
```

```
}
```

proběhne:

- A** 99x
- B** 0x
- C** kód se nepřeloží (obsahuje chybu)
- D** 100x
- E** 101x

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
```

```
for (int i = 0; i < 100; i++);
```

```
{ x -= 1; }
```

- A** -100
- B** kód se nepřeloží (obsahuje chybu)
- C** 0
- D** 1
- E** -99

- 9** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World! , Hello World!"  
**B** "!"  
**C** "Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello, Hello World!"  
**F** " , Hello World!"

- 10** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:
- A** 1  
**B** 0 pokud je A rovno 0  
**C** 0  
**D** není definováno

- 11** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 44  
**B** 1  
**C** 33  
**D** 66  
**E** 0

- 12** Datový typ int je:
- A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání celých čísel

- 13** Cyklus
- ```
int i = 0;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** 101x  
**B** 99x  
**C** kód obsahuje syntaktickou chybu  
**D** 100x

- 14** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 9 prvky s indexy 1..9 typu double  
**B** Pole s 10 prvky s indexy 1..10 typu double  
**C** Pole s 9 prvky s indexy 0..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 0..9 typu double

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		44

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

**A** 100x  
**B** 101x  
**C** kód obsahuje syntaktickou chybu  
**D** 99x
- 2** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```

**A** -100  
**B** -99  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 0  
**E** 1
- 3** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

**A** 0 pokud je A rovno 0  
**B** není definováno  
**C** 1  
**D** 0
- 4** Hodnota výrazu  $(2-20)/(5+5)$  je:

**A** 2  
**B** -1.8  
**C** -1  
**D** 0  
**E** 1
- 5** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "Hello World! , Hello World!"  
**B** "!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** " , Hello World!!"  
**E** "Hello, Hello World!!"  
**F** "Hello World!"
- 6** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 3  
**B** 6  
**C** 5  
**D** 9  
**E** kód se nepřeloží (obsahuje chybu)
- 7** Následující zápis reprezentuje:

```
int x[10];
```

**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 9 prvky s indexy 0..9 typu int  
**D** Pole s 9 prvky s indexy 1..9 typu int  
**E** Pole s 10 prvky s indexy 0..9 typu int
- 8** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

**A** 100x  
**B** 0x  
**C** 101x  
**D** 99x  
**E** kód se nepřeloží (obsahuje chybu)
- 9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
```

Bude obsah proměnné A roven:

**A** 300  
**B** 200  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** 100  
**E** kód obsahuje syntaktickou chybu

**10** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
bude obsah proměnné A roven:
```

- A** 98
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 97
- E** 101

**11** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 7 znaků
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole o 7 znacích a textový řetězec o délce 8 znaků
- D** Pole o 7 znacích a textový řetězec o délce 7 znaků
- E** Pole o 8 znacích a textový řetězec o délce 8 znaků

**12** Cyklus

```
int i = 100;
do {
    i++;
} while (i < 100);
proběhne:
```

- A** kód obsahuje syntaktickou chybu
- B** dvakrát
- C** ani jednou
- D** jednou

**13** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
Bude obsah proměnné A roven:
```

- A** 200
- B** 300
- C** 299
- D** 101
- E** kód obsahuje syntaktickou chybu
- F** 100

**14** Datový typ float je:

- A** Typ bez znaménka pro uchovávání celých čísel
- B** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- C** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
- D** Typ se znaménkem pro uchovávání celých čísel



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		45

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** není definováno
  - B** nenulová hodnota
  - C** 0 pokud je A rovno 0
  - D** 0

- 2** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** 99x
  - C** 101x
  - D** 100x

- 3** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello, Hello World!!"
  - B** "!"
  - C** "Hello World! , Hello World!!"
  - D** "Hello World! , Hello World!"
  - E** kód se nepřeloží (obsahuje chybu)
  - F** "Hello World!"

- 4** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = *A + 200;
```
- Bude obsah proměnné A roven:
- A** 100
  - B** 300
  - C** kód obsahuje syntaktickou chybu
  - D** 200
  - E** obsah hodnoty A nelze bez spuštění jednoznačně určit

- 5** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** 0
  - B** 101
  - C** -1
  - D** 1
  - E** kód se nepřeloží (obsahuje chybu)

- 6** Tělo cyklu
- ```
for (int i = 0; i <= 100; i++) {
}
```
- proběhne:
- A** 99x
  - B** 100x
  - C** 0x
  - D** 101x
  - E** kód se nepřeloží (obsahuje chybu)

- 7** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - E** Pole o 5 znacích a textový řetězec o délce 6 znaků

- 8** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** 102
  - B** 101
  - C** kód obsahuje syntaktickou chybu
  - D** 100

- 9** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 44
  - B** 0
  - C** 33
  - D** 66
  - E** 1

**10** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 1..9 typu int
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 10 prvky s indexy 0..9 typu int
- D** Pole s 10 prvky s indexy 1..10 typu int
- E** Pole s 9 prvky s indexy 0..9 typu int

**11** Cyklus

```
int i = 100;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A** ani jednou
- B** kód obsahuje syntaktickou chybu
- C** jednou
- D** dvakrát

**12** Po provedení kódu:

```
int A = 100;  
A = A++ + 200;  
Bude obsah proměnné A roven:
```

- A** 101
- B** 200
- C** 300
- D** 100
- E** 301
- F** kód obsahuje syntaktickou chybu

**13** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 6
- C** 3
- D** 9
- E** 5

**14** Datový typ int má vždy velikost:

- A** 4 bajty
- B** v závislosti na architektuře
- C** 8 bajty
- D** 2 bajty

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 46           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 5  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 6  
**D** 3  
**E** 9

- 2** Následující zápis reprezentuje:
- ```
double x[10];
```

**A** Pole s 10 prvky s indexy 1..10 typu double  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** Pole s 10 prvky s indexy 0..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 9 prvky s indexy 1..9 typu double

- 3** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "Hello World! , Hello World!"  
**B** " , Hello World!!"  
**C** "Hello, Hello World!!"  
**D** "Hello World!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "!"

- 4** Datový typ int má vždy velikost:

**A** 8 bajty  
**B** 4 bajty  
**C** v závislosti na architektuře  
**D** 2 bajty

- 5** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = pA + 200;
```

Bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 200  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** 300  
**E** 100

- 6** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** 101x  
**C** 100x  
**D** 99x

- 7** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

**A** 0  
**B** -1  
**C** 1  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101

- 8** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```

**A** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 8 znacích a textový řetězec o délce 8 znaků

- 9** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
```

Bude obsah proměnné A roven:

**A** 101  
**B** 100  
**C** 299  
**D** 200  
**E** 300  
**F** kód obsahuje syntaktickou chybu

- 10** Hodnota výrazu  $(2+3)/5+5$  je:

**A** 6  
**B** 10  
**C** 0  
**D** 1  
**E** 2

**11** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:

- A** kód se nepřeloží (obsahuje chybu)
- B** 99x
- C** 101x
- D** 0x
- E** 100x

**12** Výsledkem výrazu `(0 || A)`, kde A je proměná typu integer je:

- A** není definováno
- B** 0 pokud je A rovno 0
- C** 0
- D** 1

**13** Cyklus  
`int i = 99;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:

- A** ani jednou
- B** dvakrát
- C** jednou
- D** kód obsahuje syntaktickou chybu

**14** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
`case 99: A++; break;`  
`case 100: A++; break;`  
`case 101: A++; break;`  
`}`  
bude obsah proměné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 101
- C** 98
- D** 100
- E** 102

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		47

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2+3)/5+5$  je:

- A 1
- B 2
- C 10
- D 6
- E 0

**2** Cyklus  

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A 99x
- B kód obsahuje syntaktickou chybu
- C 101x
- D 100x

**3** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

- A 101x
- B 100x
- C 99x
- D 0x
- E kód se nepřeloží (obsahuje chybu)

**4** Datový typ char je:

- A Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- B Typ pro uchovávání malých celých čísel nebo znaků
- C Typ pro uchovávání velkých celých čísel
- D Typ bez znaménka pro uchovávání čísel s desetinnou čárkou

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 6
- B 3
- C 5
- D 9
- E kód se nepřeloží (obsahuje chybu)

**6** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměnné A roven:

- A 300
- B 100
- C kód obsahuje syntaktickou chybu
- D obsah hodnoty A nelze bez spuštění jednoznačně určit
- E 200

**7** Cyklus  

```
int i = 0;
do {
    i++;
} while (i < 100);
```

proběhne:

- A 99x
- B kód obsahuje syntaktickou chybu
- C 101x
- D 100x

**8** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 6 znacích a textový řetězec o délce 5 znaků
- B Pole o 5 znacích a textový řetězec o délce 6 znaků
- C Pole o 5 znacích a textový řetězec o délce 5 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

**9** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
```

bude obsah proměnné A roven:

- A 100
- B 101
- C 98
- D 97
- E kód obsahuje syntaktickou chybu

- 10** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```
- A** 1  
**B** 100  
**C** 99  
**D** 0  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 11** Výsledkem výrazu (1 && 0) je:
- A** 0  
**B** 1  
**C** 10  
**D** není definováno
- 
- 12** Po provedení kódu:
- ```
int A = 100;
A = A++ + 200;
```
- Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 101  
**D** 300  
**E** 301  
**F** 200
- 
- 13** Následující zápis reprezentuje:
- ```
int x[10];
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 9 prvky s indexy 1..9 typu int  
**C** Pole s 9 prvky s indexy 0..9 typu int  
**D** Pole s 10 prvky s indexy 1..10 typu int  
**E** Pole s 10 prvky s indexy 0..9 typu int
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello, Hello World!!"  
**B** "Hello World! , Hello World!"  
**C** " , Hello World!!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "!"  
**F** "Hello World!"

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		48

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměné A roven:
```

  
**A** 101  
**B** kód obsahuje syntaktickou chybu  
**C** 98  
**D** 102  
**E** 100
- 2** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:  
**A** 0  
**B** 40  
**C** 4  
**D** 60  
**E** 30
- 3** Následující zápis reprezentuje:  

```
int x[10];
```

  
**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 0..9 typu int  
**D** Pole s 9 prvky s indexy 0..9 typu int  
**E** Pole s 9 prvky s indexy 1..9 typu int
- 4** Datový typ int je:  
**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel
- 5** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** " , Hello World!!"  
**B** "Hello World!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "Hello, Hello World!!"  
**E** "!"  
**F** "Hello World! , Hello World!"
- 6** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

  
**A** 5  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 3  
**D** 9  
**E** 6
- 7** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}
```

  
proběhne:  
**A** 100x  
**B** kód obsahuje syntaktickou chybu  
**C** 99x  
**D** 101x
- 8** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
}
```

  
proběhne:  
**A** 101x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 100x  
**D** 0x  
**E** 99x
- 9** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:  
**A** 1  
**B** 10  
**C** není definováno  
**D** 0
- 10** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
    { x += 1; }
```

  
**A** 100  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 0  
**D** 99  
**E** 1

- 11** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 12** Po provedení kódu:  
`int A = 100;`  
`A = A-- + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 101
  - C** 299
  - D** 100
  - E** 300
  - F** 200
- 
- 13** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = &pA + 200;`  
Bude obsah proměnné A roven:
- A** 200
  - B** 100
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** kód obsahuje syntaktickou chybu
  - E** 300
- 
- 14** Cyklus  
`int i = 99;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** dvakrát
  - C** ani jednou
  - D** jednou



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		49

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** kód se nepřeloží (obsahuje chybu)
- B** "Hello World!"
- C** "Hello, Hello World!!"
- D** " , Hello World!!"
- E** "Hello World! , Hello World!"
- F** "!"

**2** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 100
- C** 101
- D** 102

**3** Datový typ char je:

- A** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- B** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
- C** Typ pro uchovávání malých celých čísel nebo znaků
- D** Typ pro uchovávání velkých celých čísel

**4** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 7 znaků
- B** Pole o 7 znacích a textový řetězec o délce 8 znaků
- C** Pole o 7 znacích a textový řetězec o délce 7 znaků
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole o 8 znacích a textový řetězec o délce 8 znaků

**5** Cyklus  

```
int i = 100;  
do {  
    i++;  
} while (i < 100);
```

proběhne:

- A** ani jednou
- B** dvakrát
- C** jednou
- D** kód obsahuje syntaktickou chybu

**6** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** kód obsahuje syntaktickou chybu
- C** obsah hodnoty A nelze bez spuštění jednoznačně určit
- D** 100
- E** 300

**7** Po provedení kódu:  

```
int A = 100;  
A = --A + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** 101
- C** 300
- D** kód obsahuje syntaktickou chybu
- E** 100
- F** 299

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 0
- C** 1
- D** -1
- E** 101

**9** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A** 6
- B** kód se nepřeloží (obsahuje chybu)
- C** 9
- D** 5
- E** 3

**10** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A není definováno
- B 0
- C 0 pokud je A rovno 0
- D nenulová hodnota

**11** Cyklus  

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B 100x
- C 99x
- D 101x

**12** Následující zápis reprezentuje:  

```
int x[10];
```

- A Pole s 9 prvky s indexy 0..9 typu int
- B kód se nepřeloží (obsahuje chybu)
- C Pole s 10 prvky s indexy 1..10 typu int
- D Pole s 10 prvky s indexy 0..9 typu int
- E Pole s 9 prvky s indexy 1..9 typu int

**13** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

- A kód se nepřeloží (obsahuje chybu)
- B 101x
- C 0x
- D 99x
- E 100x

**14** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 60
- B 40
- C 4
- D 30
- E 0

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		50

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  
`int A = 100;  
A = A++ + 200;`  
Bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 301  
**D** 300  
**E** 101  
**F** 200
- 2** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = *A + 200;`  
Bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 200  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** 100  
**E** 300
- 3** Cyklus  
`int i = 100;  
do {  
    i++;  
} while (i < 100);`  
proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** jednou  
**C** dvakrát  
**D** ani jednou
- 4** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**E** Pole o 5 znacích a textový řetězec o délce 5 znaků
- 5** Datový typ float je:

**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- 6** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++){  
    { x += 1; }`

**A** 1  
**B** 100  
**C** 99  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 0
- 7** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++){  
    y = y + x[i];`

**A** 5  
**B** 9  
**C** 6  
**D** 3  
**E** kód se nepřeloží (obsahuje chybu)
- 8** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");`

**A** "!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** " , Hello World!"  
**D** "Hello World!"  
**E** "Hello World! , Hello World!"  
**F** "Hello, Hello World!"
- 9** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {  
    }`  
proběhne:

**A** 101x  
**B** 100x  
**C** 99x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 0x
- 10** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

**A** 1  
**B** 0 pokud je A rovno 0  
**C** není definováno  
**D** 0

- 11** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
    `i++;`  
`}`  
proběhne:  
**A** 99x  
**B** kód obsahuje syntaktickou chybu  
**C** 101x  
**D** 100x

- 12** Následující zápis reprezentuje:  
`double x[10];`  
**A** Pole s 9 prvky s indexy 1..9 typu double  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 1..10 typu double  
**D** Pole s 9 prvky s indexy 0..9 typu double  
**E** Pole s 10 prvky s indexy 0..9 typu double

- 13** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 6  
**B** 1  
**C** 0  
**D** 2  
**E** 10

- 14** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 100: A++;`  
    `case 101: A++; break;`  
    `case 102: A++;`  
`}`  
bude obsah proměnné A roven:  
**A** 102  
**B** 101  
**C** 103  
**D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		51

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
proběhne:
```

  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 99x  
**C** 0x  
**D** 101x  
**E** 100x
- 2** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x -= 1; }
```

  
**A** -100  
**B** 1  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 0  
**E** -99
- 3** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

  
**A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 4** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

  
bude obsah proměnné A roven:  
  
**A** 102  
**B** 98  
**C** kód obsahuje syntaktickou chybu  
**D** 101  
**E** 100
- 5** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

  
**A** 9  
**B** 3  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 5  
**E** 6

- 6** Následující zápis reprezentuje:  

```
double x[10];
```

  
**A** Pole s 9 prvky s indexy 0..9 typu double  
**B** Pole s 10 prvky s indexy 0..9 typu double  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 9 prvky s indexy 1..9 typu double  
**E** Pole s 10 prvky s indexy 1..10 typu double
- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "Hello World!"  
**B** "!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "Hello, Hello World!!"  
**E** "Hello World! , Hello World!"  
**F** " , Hello World!!"
- 8** Datový typ char je:  
  
**A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**B** Typ pro uchovávání malých celých čísel nebo znaků  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ pro uchovávání velkých celých čísel
- 9** Hodnota výrazu  $(2-20)/(5+5)$  je:  
  
**A** 2  
**B** -1  
**C** 0  
**D** 1  
**E** -1.8
- 10** Po provedení kódu:  

```
int A = 100;  
A = ++A + 200;
```

  
Bude obsah proměnné A roven:  
  
**A** 301  
**B** 100  
**C** 300  
**D** kód obsahuje syntaktickou chybu  
**E** 200  
**F** 101

**11**

Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A** kód obsahuje syntaktickou chybu
- B** 99x
- C** 100x
- D** 101x

**12**

Výsledkem výrazu (1 &amp;&amp; 1) je:

- A** 1
- B** 0
- C** 10
- D** není definováno

**13**

Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = &pA + 200;
```

Bude obsah proměnné A roven:

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 200
- E** 300

**14**

Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměnná typu int proběhne:

- A** cyklus nemusí proběhnout ani jednou
- B** vždy 100x
- C** alespoň jednou
- D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		52

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus
- ```
int i = 99;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** dvakrát
  - B** jednou
  - C** ani jednou
  - D** kód obsahuje syntaktickou chybu

- 2** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** 3
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 6
  - D** 9
  - E** 5

- 3** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** 1
  - B** 0
  - C** -1
  - D** 2
  - E** -1.8

- 4** Po provedení kódu:
- ```
int A = 100;
A = ++A + 200;
```
- Bude obsah proměnné A roven:
- A** 300
  - B** 101
  - C** kód obsahuje syntaktickou chybu
  - D** 200
  - E** 100
  - F** 301

- 5** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:
- A** 0
  - B** 10
  - C** není definováno
  - D** 1

- 6** Tělo cyklu
- ```
for (int i = 1; i <= 100; i++) {
}
```
- proběhne:
- A** 99x
  - B** 100x
  - C** 101x
  - D** 0x
  - E** kód se nepřeloží (obsahuje chybu)

- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** "Hello World! , Hello World!!"
  - C** "!"
  - D** "Hello World! , Hello World!"
  - E** "Hello, Hello World!!"
  - F** "Hello World!"

- 8** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** 101
  - B** 1
  - C** 0
  - D** kód se nepřeloží (obsahuje chybu)
  - E** -1

- 9** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** 101
  - B** kód obsahuje syntaktickou chybu
  - C** 102
  - D** 100
  - E** 98

**10** Datový typ float je:

- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- B** Typ bez znaménka pro uchovávání celých čísel
- C** Typ se znaménkem pro uchovávání celých čísel
- D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**11** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = **pA + 200;
```

Bude obsah proměné A roven:

- A** 200
- B** 100
- C** 300
- D** kód obsahuje syntaktickou chybu
- E** obsah hodnoty A nelze bez spuštění jednoznačně určit

**12** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 6 znacích a textový řetězec o délce 6 znaků
- B** Pole o 6 znacích a textový řetězec o délce 5 znaků
- C** Pole o 5 znacích a textový řetězec o délce 6 znaků
- D** Pole o 5 znacích a textový řetězec o délce 5 znaků
- E** kód se nepřeloží (obsahuje chybu)

**13** Následující zápis reprezentuje:

```
double x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu double
- B** Pole s 9 prvky s indexy 1..9 typu double
- C** Pole s 9 prvky s indexy 0..9 typu double
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole s 10 prvky s indexy 1..10 typu double

**14** Cyklus

```
int i = 0;
```

```
while (i < 100) {
```

```
    i++;
```

```
}
```

proběhne:

- A** kód obsahuje syntaktickou chybu
- B** 101x
- C** 100x
- D** 99x



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 53           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
    }  
proběhne:
```

  
**A** 100x  
**B** 0x  
**C** 101x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 99x

- 2** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:
```

  
**A** 100x  
**B** 99x  
**C** 101x  
**D** kód obsahuje syntaktickou chybu

- 3** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

  
Bude obsah proměnné A roven:  
  
**A** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**B** 200  
**C** 100  
**D** kód obsahuje syntaktickou chybu  
**E** 300

- 4** Výsledkem výrazu (0 || A), kde A je proměnná typu integer je:  
  
**A** 0  
**B** 0 pokud je A rovno 0  
**C** 1  
**D** není definováno

- 5** Po provedení kódu:  

```
int A = 100;  
A = --A + 200;
```

  
Bude obsah proměnné A roven:  
  
**A** 299  
**B** 300  
**C** 200  
**D** kód obsahuje syntaktickou chybu  
**E** 100  
**F** 101

- 6** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměnné A roven:
```

  
**A** 102  
**B** 100  
**C** 101  
**D** kód obsahuje syntaktickou chybu

- 7** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

  
**A** 5  
**B** 9  
**C** 3  
**D** 6  
**E** kód se nepřeloží (obsahuje chybu)

- 8** Cyklus  

```
int i = 0;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

  
**A** 100x  
**B** 101x  
**C** kód obsahuje syntaktickou chybu  
**D** 99x

- 9** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "Hello World!"  
**B** "!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "Hello, Hello World!"  
**E** "Hello World! , Hello World!"  
**F** " , Hello World!"

**10** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 0
- B 60
- C 40
- D 30
- E 4

**11** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```

- A -99
- B 0
- C kód se nepřeloží (obsahuje chybu)
- D -100
- E 1

**12** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 8 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 8 znacích a textový řetězec o délce 7 znaků
- D Pole o 7 znacích a textový řetězec o délce 7 znaků
- E Pole o 8 znacích a textový řetězec o délce 8 znaků

**13** Datový typ int je:

- A Typ se znaménkem pro uchovávání celých čísel
- B Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- C Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D Typ bez znaménka pro uchovávání celých čísel

**14** Následující zápis reprezentuje:

```
double x[10];
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole s 9 prvky s indexy 0..9 typu double
- C Pole s 9 prvky s indexy 1..9 typu double
- D Pole s 10 prvky s indexy 0..9 typu double
- E Pole s 10 prvky s indexy 1..10 typu double

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 54           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující zápis reprezentuje:  
`double x[10];`

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole s 10 prvků s indexy 0..9 typu `double`
- C** Pole s 9 prvků s indexy 1..9 typu `double`
- D** Pole s 10 prvků s indexy 1..10 typu `double`
- E** Pole s 9 prvků s indexy 0..9 typu `double`

**2** Datový typ `float` je:

- A** Typ bez znaménka pro uchovávání celých čísel
- B** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- C** Typ se znaménkem pro uchovávání celých čísel
- D** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou

**3** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:

- A** 0x
- B** 101x
- C** 99x
- D** 100x
- E** kód se nepřeloží (obsahuje chybu)

**4** Hodnota výrazu  $(2+3)/5+5$  je:

- A** 2
- B** 10
- C** 0
- D** 6
- E** 1

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 5
- B** 6
- C** 3
- D** 9
- E** kód se nepřeloží (obsahuje chybu)

**6** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`

- A** Pole o 6 znacích a textový řetězec o délce 6 znaků
- B** Pole o 5 znacích a textový řetězec o délce 6 znaků
- C** Pole o 6 znacích a textový řetězec o délce 5 znaků
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole o 5 znacích a textový řetězec o délce 5 znaků

**7** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměnné A roven:

- A** 299
- B** 300
- C** 101
- D** kód obsahuje syntaktickou chybu
- E** 200
- F** 100

**8** Po vykonání následujícího kódu bude v `buf` uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello World!"
- B** "Hello World! , Hello World!!"
- C** "Hello, Hello World!!"
- D** "Hello World! , Hello World!"
- E** "!"
- F** kód se nepřeloží (obsahuje chybu)

**9** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A** 100
- B** 101
- C** kód obsahuje syntaktickou chybu
- D** 102

**10** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** 300
- C** obsah hodnoty A nelze bez spuštění jednoznačně určit
- D** kód obsahuje syntaktickou chybu
- E** 100

- 11** Cyklus  
`int i = 99;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:  
**A** dvakrát  
**B** kód obsahuje syntaktickou chybu  
**C** jednou  
**D** ani jednou
- 
- 12** Hodnota proměnné `x` po provedení následující kódu bude:  
`int x = 0;`  
`if (1)`  
    `x = x + 1;`  
`else`  
    `x = x - 1;`  
`for (int i = 0; i < 100; i++) {x++};`  
**A** 101  
**B** -1  
**C** 0  
**D** 1  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Výsledkem výrazu `(1 && 1)` je:  
**A** není definováno  
**B** 10  
**C** 1  
**D** 0
- 
- 14** Cyklus  
`while (i < 100) {`  
    `i++;`  
`}`  
kde `i` je proměná typu `int` proběhne:  
**A** vždy 100x  
**B** alespoň jednou  
**C** kód obsahuje syntaktickou chybu  
**D** cyklus nemusí proběhnout ani jednou

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 55           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A -1.8
- B 1
- C 0
- D -1
- E 2

**2** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = &pA + 200;
```

Bude obsah proměnné A roven:

- A 100
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C 200
- D 300
- E kód obsahuje syntaktickou chybu

**3** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++) {
    x -= 1;
}
```

- A kód se nepřeloží (obsahuje chybu)
- B -100
- C 1
- D -99
- E 0

**4** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměnné A roven:

- A 101
- B 301
- C 300
- D kód obsahuje syntaktickou chybu
- E 100
- F 200

**5** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:

- A 100x
- B 99x
- C kód se nepřeloží (obsahuje chybu)
- D 101x
- E 0x

**6** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 7 znacích a textový řetězec o délce 7 znaků
- C Pole o 8 znacích a textový řetězec o délce 7 znaků
- D Pole o 8 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 8 znaků

**7** Po provedení kódu:

```
int A = 100;
```

```
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A 103
- B kód obsahuje syntaktickou chybu
- C 102
- D 101

**8** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A není definováno
- B nenulová hodnota
- C 0
- D 0 pokud je A rovno 0

**9** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměnná typu int proběhne:

- A alespoň jednou
- B kód obsahuje syntaktickou chybu
- C vždy 100x
- D cyklus nemusí proběhnout ani jednou

**10** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu double
- B Pole s 10 prvky s indexy 0..9 typu double
- C Pole s 9 prvky s indexy 0..9 typu double
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 1..10 typu double

- 11** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```
- A** 3  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 6  
**D** 9  
**E** 5
- 
- 12** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```
- A** "Hello, Hello World!!"  
**B** "!"  
**C** "Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello World! , Hello World!!"  
**F** "Hello World! , Hello World!"
- 
- 13** Cyklus
- ```
do {  
    i++;  
} while (i < 100);
```
- kde i je proměnná typu int proběhne:
- A** alespoň jednou  
**B** kód obsahuje syntaktickou chybu  
**C** cyklus nemusí proběhnout ani jednou  
**D** vždy 100x
- 
- 14** Datový typ int je:
- A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání celých čísel

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		56

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A 0
- B nenulová hodnota
- C 0 pokud je A rovno 0
- D není definováno

**2** Datový typ float je:

- A Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- B Typ se znaménkem pro uchovávání celých čísel
- C Typ bez znaménka pro uchovávání celých čísel
- D Typ bez znaménka pro uchovávání čísel s desetinnou čárkou

**3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A 5
- B 6
- C 3
- D 9
- E kód se nepřeloží (obsahuje chybu)

**4** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A 102
- B kód obsahuje syntaktickou chybu
- C 100
- D 101
- E 98

**5** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A "Hello World!"
- B "Hello, Hello World!!"
- C kód se nepřeloží (obsahuje chybu)
- D " , Hello World!!"
- E "Hello World! , Hello World!"
- F "!"

**6** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A 100x
- B 99x
- C 101x
- D kód obsahuje syntaktickou chybu

**7** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 6 znacích a textový řetězec o délce 5 znaků
- C Pole o 5 znacích a textový řetězec o délce 6 znaků
- D Pole o 5 znacích a textový řetězec o délce 5 znaků
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

**8** Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

- A dvakrát
- B ani jednou
- C jednou
- D kód obsahuje syntaktickou chybu

**9** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

- A -100
- B 1
- C -99
- D kód se nepřeloží (obsahuje chybu)
- E 0

**10** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 1
- B 44
- C 0
- D 33
- E 66

**11** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu int
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 10 prvky s indexy 1..10 typu int
- D** Pole s 9 prvky s indexy 0..9 typu int
- E** Pole s 9 prvky s indexy 1..9 typu int

**12** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = *pA + 200;
```

Bude obsah proměnné A roven:

- A** 100
- B** kód obsahuje syntaktickou chybu
- C** obsah hodnoty A nelze bez spuštění jednoznačně určit
- D** 200
- E** 300

**13** Po provedení kódu:

```
int A = 100;
```

```
A = --A + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** 100
- C** kód obsahuje syntaktickou chybu
- D** 101
- E** 300
- F** 299

**14** Tělo cyklu

```
for (int i = 0; i < 100; i++) {  
    }
```

proběhne:

- A** 101x
- B** 0x
- C** 100x
- D** kód se nepřeloží (obsahuje chybu)
- E** 99x



# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		57

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 0
- B -1
- C 2
- D -1.8
- E 1

**2** Následující zápis reprezentuje:  
`int x[10];`

- A kód se nepřeloží (obsahuje chybu)
- B Pole s 10 prvky s indexy 0..9 typu int
- C Pole s 9 prvky s indexy 0..9 typu int
- D Pole s 9 prvky s indexy 1..9 typu int
- E Pole s 10 prvky s indexy 1..10 typu int

**3** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
    `i++;`  
`}`  
proběhne:

- A 101x
- B 99x
- C kód obsahuje syntaktickou chybu
- D 100x

**4** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {`  
    `}`  
proběhne:

- A 99x
- B 100x
- C kód se nepřeloží (obsahuje chybu)
- D 101x
- E 0x

**5** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcat(buf, " , Hello World!");`  
`strcat(buf, "!");`

- A "Hello World!"
- B "!"
- C "Hello World! , Hello World!"
- D kód se nepřeloží (obsahuje chybu)
- E "Hello World! , Hello World!!"
- F "Hello, Hello World!!"

**6** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:

- A 1
- B 0 pokud je A rovno 0
- C není definováno
- D 0

**7** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 6
- B 9
- C 3
- D 5
- E kód se nepřeloží (obsahuje chybu)

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A -1
- B 101
- C kód se nepřeloží (obsahuje chybu)
- D 1
- E 0

**9** Datový typ float je:

- A Typ se znaménkem pro uchovávání celých čísel
- B Typ bez znaménka pro uchovávání celých čísel
- C Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**10** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 7 znaků
- B Pole o 8 znacích a textový řetězec o délce 8 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 7 znacích a textový řetězec o délce 8 znaků
- E Pole o 8 znacích a textový řetězec o délce 7 znaků

- 11** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = **pA + 200;`  
Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** kód obsahuje syntaktickou chybu
  - C** 300
  - D** 200
  - E** 100

- 12** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
    `default: A--;`  
`}`  
bude obsah proměnné A roven:
- A** 101
  - B** 100
  - C** kód obsahuje syntaktickou chybu
  - D** 98
  - E** 97

- 13** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 100
  - B** 101
  - C** kód obsahuje syntaktickou chybu
  - D** 200
  - E** 300
  - F** 301

- 14** Cyklus  
`do {`  
    `i++;`  
`} while (i < 100);`  
kde i je proměnná typu int proběhne:
- A** alespoň jednou
  - B** vždy 100x
  - C** cyklus nemusí proběhnout ani jednou
  - D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		58

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 3
- C 9
- D 5
- E 6

- 2** Cyklus

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A 100x
- B 99x
- C kód obsahuje syntaktickou chybu
- D 101x

- 3** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 7 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 7 znacích a textový řetězec o délce 8 znaků
- D Pole o 7 znacích a textový řetězec o délce 7 znaků
- E Pole o 8 znacích a textový řetězec o délce 8 znaků

- 4** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
```

Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 300
- C 100
- D 200
- E obsah hodnoty A nelze bez spuštění jednoznačně určit

- 5** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "Hello World!"
- C " , Hello World!!"
- D "!"
- E "Hello, Hello World!!"
- F "Hello World! , Hello World!"

- 6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A 0
- B 1
- C 101
- D -1
- E kód se nepřeloží (obsahuje chybu)

- 7** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
```

bude obsah proměnné A roven:

- A 98
- B kód obsahuje syntaktickou chybu
- C 101
- D 100
- E 97

- 8** Datový typ int je:

- A Typ se znaménkem pro uchovávání celých čísel
- B Typ bez znaménka pro uchovávání celých čísel
- C Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
- D Typ se znaménkem pro uchovávání čísel s desetinnou čárkou

- 9** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {`  
`}`  
proběhne:  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 100x  
**C** 101x  
**D** 0x  
**E** 99x
- 
- 10** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměná typu integer je:  
**A** 0 pokud je A rovno 0  
**B** není definováno  
**C** 0  
**D** nenulová hodnota
- 
- 11** Následující zápis reprezentuje:  
`int x[10];`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 9 prvky s indexy 1..9 typu int  
**C** Pole s 10 prvky s indexy 1..10 typu int  
**D** Pole s 10 prvky s indexy 0..9 typu int  
**E** Pole s 9 prvky s indexy 0..9 typu int
- 
- 12** Cyklus  
`do {`  
`i++;`  
`} while (i < 100);`  
kde i je proměná typu int proběhne:  
**A** vždy 100x  
**B** cyklus nemusí proběhnout ani jednou  
**C** kód obsahuje syntaktickou chybu  
**D** alespoň jednou
- 
- 13** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 0  
**B** 1  
**C** 10  
**D** 6  
**E** 2
- 
- 14** Po provedení kódu:  
`int A = 100;`  
`A = --A + 200;`  
Bude obsah proměné A roven:  
**A** 100  
**B** 200  
**C** 101  
**D** 299  
**E** kód obsahuje syntaktickou chybu  
**F** 300

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		59

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** 0 pokud je A rovno 0
  - B** 0
  - C** není definováno
  - D** nenulová hodnota

- 2** Cyklus
- ```
int i = 100;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** dvakrát
  - B** ani jednou
  - C** kód obsahuje syntaktickou chybu
  - D** jednou

- 3** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 9 prvky s indexy 0..9 typu double
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole s 10 prvky s indexy 0..9 typu double
  - D** Pole s 10 prvky s indexy 1..10 typu double
  - E** Pole s 9 prvky s indexy 1..9 typu double

- 4** Tělo cyklu
- ```
for (int i = 1; i <= 100; i++) {
}
```
- proběhne:
- A** 0x
  - B** 100x
  - C** 99x
  - D** 101x
  - E** kód se nepřeloží (obsahuje chybu)

- 5** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** 5
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 9
  - D** 6
  - E** 3

- 6** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```
- bude obsah proměnné A roven:
- A** 103
  - B** kód obsahuje syntaktickou chybu
  - C** 101
  - D** 102

- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World!"
  - B** "Hello World! , Hello World!"
  - C** "Hello, Hello World!!"
  - D** "!"
  - E** kód se nepřeloží (obsahuje chybu)
  - F** " , Hello World!!"

- 8** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** -1
  - B** 1
  - C** 0
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 101

- 9** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** 100x
  - B** kód obsahuje syntaktickou chybu
  - C** 99x
  - D** 101x

- 10** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = **pA + 200;`  
Bude obsah proměnné A roven:  
**A** 200  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** 100  
**E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 
- 11** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 300  
**D** 301  
**E** 101  
**F** 200
- 
- 12** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`  
**A** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků
- 
- 13** Datový typ float je:  
**A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**B** Typ se znaménkem pro uchovávání celých čísel  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání celých čísel
- 
- 14** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 0  
**B** 44  
**C** 1  
**D** 66  
**E** 33

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		60

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Výsledkem výrazu (1 && 1) je:

- A 1
- B 10
- C 0
- D není definováno

**2** Následující zápis reprezentuje:  
`int x[10];`

- A Pole s 10 prvky s indexy 1..10 typu int
- B kód se nepřeloží (obsahuje chybu)
- C Pole s 10 prvky s indexy 0..9 typu int
- D Pole s 9 prvky s indexy 1..9 typu int
- E Pole s 9 prvky s indexy 0..9 typu int

**3** Cyklus  
`while (i < 100) {  
    i++;  
}`

kde i je proměná typu int proběhne:

- A cyklus nemusí proběhnout ani jednou
- B kód obsahuje syntaktickou chybu
- C alespoň jednou
- D vždy 100x

**4** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "!"
- B "Hello World!"
- C "Hello, Hello World!"
- D " , Hello World!!"
- E "Hello World! , Hello World!"
- F kód se nepřeloží (obsahuje chybu)

**5** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *A + 200;  
Bude obsah proměné A roven:
```

- A obsah hodnoty A nelze bez spuštění jednoznačně určit
- B 200
- C 300
- D 100
- E kód obsahuje syntaktickou chybu

**6** Po provedení kódu:

```
int A = 100;  
A = --A + 200;  
Bude obsah proměné A roven:
```

- A 100
- B 300
- C kód obsahuje syntaktickou chybu
- D 101
- E 200
- F 299

**7** Tělo cyklu  
`for (int i = 0; i < 100; i++) {  
}`  
proběhne:

- A 99x
- B kód se nepřeloží (obsahuje chybu)
- C 0x
- D 100x
- E 101x

**8** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`

- A Pole o 8 znacích a textový řetězec o délce 8 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 7 znacích a textový řetězec o délce 8 znaků
- D Pole o 7 znacích a textový řetězec o délce 7 znaků
- E Pole o 8 znacích a textový řetězec o délce 7 znaků

**9** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}  
bude obsah proměné A roven:
```

- A 100
- B 98
- C 97
- D kód obsahuje syntaktickou chybu
- E 101

**10** Hodnota výrazu (2+3)/5+5 je:

- A 2
- B 6
- C 1
- D 10
- E 0

**11** Datový typ int má vždy velikost:

- A v závislosti na architektuře
- B 8 bajty
- C 4 bajty
- D 2 bajty

**12** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 5
- B 6
- C 3
- D 9
- E kód se nepřeloží (obsahuje chybu)

**13** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A 1
- B kód se nepřeloží (obsahuje chybu)
- C 101
- D 0
- E -1

**14** Cyklus

```
int i = 0;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A kód obsahuje syntaktickou chybu
- B 100x
- C 99x
- D 101x



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		61

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
}
```

  
proběhne:  
**A** 99x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 101x  
**D** 0x  
**E** 100x

- 2** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

  
**A** 3  
**B** 9  
**C** 6  
**D** 5  
**E** kód se nepřeloží (obsahuje chybu)

- 3** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 44  
**B** 1  
**C** 33  
**D** 66  
**E** 0

- 4** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}
```

  
proběhne:  
**A** 99x  
**B** kód obsahuje syntaktickou chybu  
**C** 100x  
**D** 101x

- 5** Datový typ int má vždy velikost:  
**A** 4 bajty  
**B** 2 bajty  
**C** 8 bajty  
**D** v závislosti na architektuře

- 6** Cyklus  

```
int i = 0;  
do {  
    i++;  
} while (i < 100);
```

  
proběhne:  
**A** 99x  
**B** 100x  
**C** kód obsahuje syntaktickou chybu  
**D** 101x

- 7** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

  
bude obsah proměnné A roven:  
**A** 102  
**B** kód obsahuje syntaktickou chybu  
**C** 101  
**D** 100  
**E** 98

- 8** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:  
**A** 0 pokud je A rovno 0  
**B** 0  
**C** 1  
**D** není definováno

- 9** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

  
Bude obsah proměnné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 300  
**C** 200  
**D** 100  
**E** obsah hodnoty A nelze bez spuštění jednoznačně určit

- 10** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");`
- A** "Hello, Hello World!!"  
**B** "!"  
**C** "Hello World!"  
**D** "Hello World! , Hello World!"  
**E** "Hello World! , Hello World!"  
**F** kód se nepřeloží (obsahuje chybu)
- 
- 11** Po provedení kódu:  
`int A = 100;  
A = --A + 200;`  
Bude obsah proměnné A roven:
- A** 100  
**B** 300  
**C** 101  
**D** kód obsahuje syntaktickou chybu  
**E** 200  
**F** 299
- 
- 12** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++);  
{ x -= 1; }`
- A** 1  
**B** -100  
**C** 0  
**D** -99  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 9 prvky s indexy 0..9 typu double  
**B** Pole s 10 prvky s indexy 0..9 typu double  
**C** Pole s 9 prvky s indexy 1..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 1..10 typu double
- 
- 14** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		62

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 2
- B -1.8
- C 0
- D 1
- E -1

**2** Následující zápis reprezentuje:  
`double x[10];`

- A Pole s 9 prvky s indexy 1..9 typu double
- B Pole s 10 prvky s indexy 0..9 typu double
- C Pole s 9 prvky s indexy 0..9 typu double
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 1..10 typu double

**3** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A 1
- B kód se nepřeloží (obsahuje chybu)
- C -1
- D 101
- E 0

**4** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`

- A Pole o 8 znacích a textový řetězec o délce 7 znaků
- B Pole o 8 znacích a textový řetězec o délce 8 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 7 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 7 znaků

**5** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A 1
- B není definováno
- C 10
- D 0

**6** Cyklus  
`int i = 0;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:

- A 100x
- B 99x
- C 101x
- D kód obsahuje syntaktickou chybu

**7** Datový typ int má vždy velikost:

- A 8 bajty
- B 2 bajty
- C 4 bajty
- D v závislosti na architektuře

**8** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 100
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 300
- E 200

**9** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
    `}`  
proběhne:

- A 99x
- B 101x
- C 0x
- D 100x
- E kód se nepřeloží (obsahuje chybu)

**10** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A "!"
- B "Hello, Hello World!!"
- C "Hello World!"
- D "Hello World! , Hello World!"
- E kód se nepřeloží (obsahuje chybu)
- F " , Hello World!!"

**11** Po provedení kódu:

```
int A = 100;
A = A++ + 200;
```

Bude obsah proměnné A roven:

- A 101
- B 100
- C kód obsahuje syntaktickou chybu
- D 200
- E 301
- F 300

- 12** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 100
  - C** 101
  - D** 102

- 13** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** 100x
  - B** 101x
  - C** kód obsahuje syntaktickou chybu
  - D** 99x

- 14** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 6
  - B** 9
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 3
  - E** 5

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 63           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World!"
- B "Hello World! , Hello World!!"
- C "!"
- D "Hello, Hello World!!"
- E kód se nepřeloží (obsahuje chybu)
- F "Hello World! , Hello World!"

**2** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A -1
- B 1
- C 101
- D kód se nepřeloží (obsahuje chybu)
- E 0

**3** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A 0
- B 0 pokud je A rovno 0
- C nenulová hodnota
- D není definováno

**4** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 60
- B 40
- C 0
- D 30
- E 4

**5** Po provedení kódu:

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}
```

bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 103
- C 101
- D 102

**6** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

  
proběhne:

- A 100x
- B 101x
- C 99x
- D kód obsahuje syntaktickou chybu

**7** Následující zápis reprezentuje:  

```
double x[10];
```

- A Pole s 9 prvky s indexy 0..9 typu double
- B Pole s 10 prvky s indexy 0..9 typu double
- C Pole s 10 prvky s indexy 1..10 typu double
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 9 prvky s indexy 1..9 typu double

**8** Datový typ int má vždy velikost:

- A v závislosti na architektuře
- B 2 bajty
- C 4 bajty
- D 8 bajty

**9** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
}
```

  
proběhne:

- A 100x
- B 101x
- C 0x
- D 99x
- E kód se nepřeloží (obsahuje chybu)

**10** Cyklus  

```
do {  
    i++;  
} while (i < 100);
```

  
kde i je proměnná typu int proběhne:

- A kód obsahuje syntaktickou chybu
- B vždy 100x
- C alespoň jednou
- D cyklus nemusí proběhnout ani jednou

- 11** Po provedení kódu:  
`int A = 100;`  
`A = A-- + 200;`  
Bude obsah proměnné A roven:
- A** 200
  - B** 101
  - C** 299
  - D** 300
  - E** 100
  - F** kód obsahuje syntaktickou chybu
- 
- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = &pA + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 200
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 100
  - E** 300
- 
- 13** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - E** Pole o 6 znacích a textový řetězec o délce 6 znaků
- 
- 14** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i < 3; i++)`  
`y = y + x[i];`
- A** 6
  - B** 3
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 5
  - E** 9

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 64           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = pA + 200;
```

Bude obsah proměné A roven:

**A** 300  
**B** 100  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** 200  
**E** kód obsahuje syntaktickou chybu
- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "Hello World!"  
**B** "!"  
**C** "Hello, Hello World!!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** " , Hello World!!"  
**F** "Hello World! , Hello World!"
- 3** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```

}  
bude obsah proměné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 98  
**C** 100  
**D** 97  
**E** 101
- 4** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** -1  
**C** 101  
**D** 1  
**E** 0
- 5** Výsledkem výrazu (1 && 0) je:

**A** 1  
**B** 0  
**C** 10  
**D** není definováno
- 6** Následující zápis reprezentuje:

```
double x[10];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** Pole s 10 prvky s indexy 0..9 typu double  
**D** Pole s 10 prvky s indexy 1..10 typu double  
**E** Pole s 9 prvky s indexy 1..9 typu double
- 7** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměné A roven:

**A** 101  
**B** 100  
**C** 299  
**D** 200  
**E** 300  
**F** kód obsahuje syntaktickou chybu
- 8** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 6 znaků
- 9** Tělo cyklu

```
for (int i = 0; i < 100; i++) {
}
```

proběhne:

**A** 100x  
**B** 101x  
**C** 99x  
**D** 0x  
**E** kód se nepřeloží (obsahuje chybu)
- 10** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměná typu int proběhne:

**A** cyklus nemusí proběhnout ani jednou  
**B** alespoň jednou  
**C** vždy 100x  
**D** kód obsahuje syntaktickou chybu

- 11** Cyklus
- ```
int i = 0;
while (i < 100) {
    i++;
}
```
- proběhne:
- A** 101x
  - B** 99x
  - C** kód obsahuje syntaktickou chybu
  - D** 100x

- 12** Datový typ int má vždy velikost:
- A** 2 bajty
  - B** v závislosti na architektuře
  - C** 4 bajty
  - D** 8 bajty

- 13** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 5
  - B** 6
  - C** 3
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 9

- 14** Hodnota výrazu  $(2+3)/5+5$  je:
- A** 10
  - B** 6
  - C** 2
  - D** 1
  - E** 0



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 65           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 99x  
**D** 101x
- 2** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

**A** 99x  
**B** 100x  
**C** 0x  
**D** 101x  
**E** kód se nepřeloží (obsahuje chybu)
- 3** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 103  
**C** 102  
**D** 101
- 4** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměné A roven:

**A** 101  
**B** 301  
**C** 300  
**D** kód obsahuje syntaktickou chybu  
**E** 200  
**F** 100
- 5** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

**A** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 5 znacích a textový řetězec o délce 5 znaků
- 6** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

**A** 6  
**B** 3  
**C** 9  
**D** 5  
**E** kód se nepřeloží (obsahuje chybu)
- 7** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:

**A** není definováno  
**B** 0 pokud je A rovno 0  
**C** 1  
**D** 0
- 8** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměná typu int proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** vždy 100x  
**C** alespoň jednou  
**D** cyklus nemusí proběhnout ani jednou
- 9** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

**A** 0  
**B** kód se nepřeloží (obsahuje chybu)  
**C** -99  
**D** -100  
**E** 1
- 10** Následující zápis reprezentuje:

```
double x[10];
```

**A** Pole s 9 prvky s indexy 0..9 typu double  
**B** Pole s 9 prvky s indexy 1..9 typu double  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** Pole s 10 prvky s indexy 1..10 typu double
- 11** Datový typ int má vždy velikost:

**A** 8 bajty  
**B** 2 bajty  
**C** 4 bajty  
**D** v závislosti na architektuře

**12** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** "Hello, Hello World!!"
- B** "!"
- C** kód se nepřeloží (obsahuje chybu)
- D** " , Hello World!!"
- E** "Hello World! , Hello World!"
- F** "Hello World!"

**13** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;  
Bude obsah proměnné A roven:
```

- A** kód obsahuje syntaktickou chybu
- B** obsah hodnoty A nelze bez spuštění jednoznačně určit
- C** 300
- D** 100
- E** 200

**14** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A** 0
- B** 33
- C** 1
- D** 66
- E** 44

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 66           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Výsledkem výrazu (1 && 0) je:

- A 10
- B není definováno
- C 1
- D 0

**2** Hodnota výrazu (2+3)/5+5 je:

- A 6
- B 0
- C 2
- D 1
- E 10

**3** Následující zápis reprezentuje:  
double x[10];

- A kód se nepřeloží (obsahuje chybu)
- B Pole s 10 prvky s indexy 0..9 typu double
- C Pole s 9 prvky s indexy 0..9 typu double
- D Pole s 10 prvky s indexy 1..10 typu double
- E Pole s 9 prvky s indexy 1..9 typu double

**4** Cyklus  
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:

- A kód obsahuje syntaktickou chybu
- B 101x
- C 99x
- D 100x

**5** Tělo cyklu  
for (int i = 0; i <= 100; i++) {  
}  
proběhne:

- A kód se nepřeloží (obsahuje chybu)
- B 100x
- C 0x
- D 99x
- E 101x

**6** Cyklus  
int i = 99;  
do {  
    i++;  
} while (i < 100);  
proběhne:

- A ani jednou
- B jednou
- C dvakrát
- D kód obsahuje syntaktickou chybu

**7** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A 0
- B kód se nepřeloží (obsahuje chybu)
- C 1
- D -1
- E 101

**8** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 6
- B 3
- C 9
- D kód se nepřeloží (obsahuje chybu)
- E 5

**9** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *A + 200;  
Bude obsah proměnné A roven:
```

- A 100
- B 200
- C kód obsahuje syntaktickou chybu
- D obsah hodnoty A nelze bez spuštění jednoznačně určit
- E 300

**10** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}
```

bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 101
- C 98
- D 97
- E 100

- 11** Datový typ int je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - D** Typ se znaménkem pro uchovávání celých čísel

- 12** Po provedení kódu:  
`int A = 100;`  
`A = A-- + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 300
  - C** 100
  - D** kód obsahuje syntaktickou chybu
  - E** 299
  - F** 200

- 13** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - E** Pole o 5 znacích a textový řetězec o délce 5 znaků

- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcat(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** "Hello World! , Hello World!!"
  - B** "Hello World!"
  - C** "!"
  - D** kód se nepřeloží (obsahuje chybu)
  - E** "Hello World! , Hello World!"
  - F** "Hello, Hello World!!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 67           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus
- ```
int i = 99;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** dvakrát
  - B** kód obsahuje syntaktickou chybu
  - C** ani jednou
  - D** jednou

- 2** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```
- A** 100
  - B** 1
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 99
  - E** 0

- 3** Tělo cyklu
- ```
for (int i = 1; i < 100; i++) {
}
```
- proběhne:
- A** kód se nepřeloží (obsahuje chybu)
  - B** 0x
  - C** 99x
  - D** 100x
  - E** 101x

- 4** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 10 prvky s indexy 1..10 typu double
  - B** Pole s 10 prvky s indexy 0..9 typu double
  - C** Pole s 9 prvky s indexy 0..9 typu double
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole s 9 prvky s indexy 1..9 typu double

- 5** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** 101x
  - B** kód obsahuje syntaktickou chybu
  - C** 100x
  - D** 99x

- 6** Výsledkem výrazu (1 && 1) je:
- A** 1
  - B** není definováno
  - C** 10
  - D** 0

- 7** Po provedení kódu:
- ```
int A = 100;
A = --A + 200;
```
- Bude obsah proměnné A roven:
- A** 299
  - B** 101
  - C** 300
  - D** 100
  - E** kód obsahuje syntaktickou chybu
  - F** 200

- 8** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 6
  - B** 3
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 5
  - E** 9

- 9** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "!"
  - B** "Hello World! , Hello World!"
  - C** " , Hello World!!"
  - D** kód se nepřeloží (obsahuje chybu)
  - E** "Hello, Hello World!!"
  - F** "Hello World!"

- 10** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```
- A** Pole o 7 znacích a textový řetězec o délce 8 znaků
  - B** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - C** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**11** Datový typ `int` má vždy velikost:

- A** 2 bajty
- B** 4 bajty
- C** v závislosti na architektuře
- D** 8 bajty

**12** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A** -1
- B** 2
- C** 1
- D** -1.8
- E** 0

**13** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```

bude obsah proměnné `A` roven:

- A** 97
- B** 101
- C** 98
- D** kód obsahuje syntaktickou chybu
- E** 100

**14** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = &pA + 200;
```

Bude obsah proměnné `A` roven:

- A** kód obsahuje syntaktickou chybu
- B** 200
- C** 300
- D** 100
- E** obsah hodnoty `A` nelze bez spuštění jednoznačně určit

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		68

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
proběhne:
```

  
**A** 100x  
**B** 99x  
**C** 0x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101x

- 2** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 1  
**C** 101  
**D** -1  
**E** 0

- 3** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

  
**A** 9  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 6  
**D** 5  
**E** 3

- 4** Datový typ int je:  
**A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou  
**C** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou  
**D** Typ bez znaménka pro uchovávání celých čísel

- 5** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;
```

  
bude obsah proměnné A roven:  
**A** 98  
**B** 100  
**C** kód obsahuje syntaktickou chybu  
**D** 97  
**E** 101

- 6** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;
```

  
Bude obsah proměnné A roven:  
**A** 100  
**B** 300  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** kód obsahuje syntaktickou chybu  
**E** 200

- 7** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

  
kde i je proměnná typu int proběhne:  
**A** alespoň jednou  
**B** kód obsahuje syntaktickou chybu  
**C** vždy 100x  
**D** cyklus nemusí proběhnout ani jednou

- 8** Následující zápis reprezentuje:  

```
int x[10];
```

  
**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 0..9 typu int  
**D** Pole s 9 prvky s indexy 0..9 typu int  
**E** Pole s 9 prvky s indexy 1..9 typu int

- 9** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 6  
**B** 10  
**C** 1  
**D** 2  
**E** 0

- 10** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!!");`
- A** "Hello World! , Hello World!"  
**B** "!"  
**C** "Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello, Hello World!!"  
**F** "Hello World! , Hello World!!"
- 
- 11** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**E** Pole o 8 znacích a textový řetězec o délce 7 znaků
- 
- 12** Cyklus  
`int i = 0;  
do {  
 i++;  
} while (i < 100);`  
proběhne:
- A** 100x  
**B** 101x  
**C** kód obsahuje syntaktickou chybu  
**D** 99x
- 
- 13** Po provedení kódu:  
`int A = 100;  
A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 101  
**C** 100  
**D** 301  
**E** 200  
**F** 300
- 
- 14** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:
- A** 0 pokud je A rovno 0  
**B** 1  
**C** není definováno  
**D** 0



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		69

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** " , Hello World!!"  
**B** "Hello World! , Hello World!"  
**C** "Hello, Hello World!!"  
**D** "!"  
**E** "Hello World!"  
**F** kód se nepřeloží (obsahuje chybu)

- 2** Výsledkem výrazu (1 && 0) je:
- A** 0  
**B** není definováno  
**C** 1  
**D** 10

- 3** Tělo cyklu
- ```
for (int i = 0; i <= 100; i++) {
}
```
- proběhne:
- A** 100x  
**B** 99x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 0x  
**E** 101x

- 4** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** -99  
**C** 0  
**D** 1  
**E** -100

- 5** Datový typ float je:
- A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou

- 6** Po provedení kódu:
- ```
int A = 100;
A = ++A + 200;
```
- Bude obsah proměnné A roven:
- A** 301  
**B** 100  
**C** kód obsahuje syntaktickou chybu  
**D** 101  
**E** 300  
**F** 200

- 7** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = &pA + 200;
```
- Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 200  
**C** 100  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 300

- 8** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** 3  
**C** 9  
**D** 6  
**E** 5

- 9** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```
- }
- bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 101  
**D** 97  
**E** 98

- 10** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - B** Pole o 8 znacích a textový řetězec o délce 7 znaků
  - C** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - D** Pole o 7 znacích a textový řetězec o délce 8 znaků
  - E** kód se nepřeloží (obsahuje chybu)

- 11** Cyklus  
`int i = 99;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:
- A** ani jednou
  - B** kód obsahuje syntaktickou chybu
  - C** jednou
  - D** dvakrát

- 12** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 1
  - B** 33
  - C** 44
  - D** 0
  - E** 66

- 13** Cyklus  
`while (i < 100) {`  
    `i++;`  
`}`  
kde `i` je proměnná typu `int` proběhne:
- A** alespoň jednou
  - B** vždy 100x
  - C** kód obsahuje syntaktickou chybu
  - D** cyklus nemusí proběhnout ani jednou

- 14** Následující zápis reprezentuje:  
`double x[10];`
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole s 10 prvky s indexy 0..9 typu `double`
  - C** Pole s 9 prvky s indexy 1..9 typu `double`
  - D** Pole s 9 prvky s indexy 0..9 typu `double`
  - E** Pole s 10 prvky s indexy 1..10 typu `double`

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 70           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!!");
```

- A "Hello World! , Hello World!"
- B "!"
- C "Hello, Hello World!!"
- D kód se nepřeloží (obsahuje chybu)
- E "Hello World!"
- F "Hello World! , Hello World!!"

**2** Následující zápis reprezentuje:  

```
int x[10];
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole s 9 prvky s indexy 0..9 typu int
- C Pole s 10 prvky s indexy 0..9 typu int
- D Pole s 10 prvky s indexy 1..10 typu int
- E Pole s 9 prvky s indexy 1..9 typu int

**3** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;  
Bude obsah proměnné A roven:
```

- A kód obsahuje syntaktickou chybu
- B 300
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 200
- E 100

**4** Datový typ float je:

- A Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ se znaménkem pro uchovávání celých čísel
- D Typ bez znaménka pro uchovávání celých čísel

**5** Po provedení kódu:

```
int A = 100;  
A = A++ + 200;  
Bude obsah proměnné A roven:
```

- A 300
- B 301
- C 100
- D kód obsahuje syntaktickou chybu
- E 101
- F 200

**6** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 1
- B 2
- C -1
- D 0
- E -1.8

**7** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A není definováno
- B 10
- C 1
- D 0

**8** Cyklus

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

proběhne:

- A 100x
- B 101x
- C kód obsahuje syntaktickou chybu
- D 99x

**9** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A kód se nepřeloží (obsahuje chybu)
- B 101
- C -1
- D 0
- E 1

**10** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 6 znacích a textový řetězec o délce 5 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 5 znacích a textový řetězec o délce 5 znaků
- D Pole o 6 znacích a textový řetězec o délce 6 znaků
- E Pole o 5 znacích a textový řetězec o délce 6 znaků

- 11** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** cyklus nemusí proběhnout ani jednou  
**B** kód obsahuje syntaktickou chybu  
**C** vždy 100x  
**D** alespoň jednou

- 12** Tělo cyklu  
for (int i = 0; i <= 100; i++) {  
    }  
proběhne:  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 101x  
**C** 0x  
**D** 99x  
**E** 100x

- 13** Po provedení kódu:  
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměné A roven:  
**A** 102  
**B** 101  
**C** kód obsahuje syntaktickou chybu  
**D** 100

- 14** Hodnota proměnné po provedení následujícího kódu bude:  
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];  
**A** 9  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 3  
**D** 5  
**E** 6

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 71           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Datový typ int je:

- A** Typ bez znaménka pro uchovávání celých čísel
- B** Typ se znaménkem pro uchovávání celých čísel
- C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**2** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A** kód se nepřeloží (obsahuje chybu)
- B** "Hello World!"
- C** "Hello, Hello World!!"
- D** " , Hello World!!"
- E** "!"
- F** "Hello World! , Hello World!"

**3** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
Bude obsah proměně A roven:
```

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 200
- E** 300

**4** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
bude obsah proměně A roven:
```

- A** 100
- B** 102
- C** 98
- D** kód obsahuje syntaktickou chybu
- E** 101

**5** Výsledkem výrazu (1 && 1) je:

- A** 1
- B** 0
- C** 10
- D** není definováno

**6** Po provedení kódu:

```
int A = 100;
A = --A + 200;
Bude obsah proměně A roven:
```

- A** 200
- B** 100
- C** 300
- D** kód obsahuje syntaktickou chybu
- E** 101
- F** 299

**7** Hodnota výrazu (2/3)\*(10+11+12)\*2 je:

- A** 0
- B** 33
- C** 44
- D** 1
- E** 66

**8** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {
}
proběhne:
```

- A** 100x
- B** 99x
- C** 0x
- D** 101x
- E** kód se nepřeloží (obsahuje chybu)

**9** Následující zápis reprezentuje:

```
double x[10];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole s 9 prvky s indexy 0..9 typu double
- C** Pole s 10 prvky s indexy 1..10 typu double
- D** Pole s 10 prvky s indexy 0..9 typu double
- E** Pole s 9 prvky s indexy 1..9 typu double

**10** Cyklus

```
int i = 0;
while (i < 100) {
    i++;
}
proběhne:
```

- A** 99x
- B** 100x
- C** 101x
- D** kód obsahuje syntaktickou chybu

**11** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 5 znacích a textový řetězec o délce 6 znaků
- B** Pole o 5 znacích a textový řetězec o délce 5 znaků
- C** Pole o 6 znacích a textový řetězec o délce 5 znaků
- D** Pole o 6 znacích a textový řetězec o délce 6 znaků
- E** kód se nepřeloží (obsahuje chybu)

- 12** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** 0  
**C** 1  
**D** 100  
**E** 99
- 
- 13** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** 6  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 3  
**D** 5  
**E** 9
- 
- 14** Cyklus
- ```
int i = 99;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** jednou  
**B** dvakrát  
**C** ani jednou  
**D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		72

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```
- bude obsah proměnné A roven:
- A** 98
  - B** 101
  - C** kód obsahuje syntaktickou chybu
  - D** 100
  - E** 97

- 2** Výsledkem výrazu (1 && 0) je:
- A** 0
  - B** 10
  - C** není definováno
  - D** 1

- 3** Datový typ int má vždy velikost:
- A** 2 bajty
  - B** 4 bajty
  - C** 8 bajty
  - D** v závislosti na architektuře

- 4** Tělo cyklu
- ```
for (int i = 0; i < 100; i++) {
}
```
- proběhne:
- A** 101x
  - B** 0x
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 100x
  - E** 99x

- 5** Po provedení kódu:
- ```
int A = 100;
A = ++A + 200;
```
- Bude obsah proměnné A roven:
- A** 300
  - B** 101
  - C** 200
  - D** kód obsahuje syntaktickou chybu
  - E** 100
  - F** 301

- 6** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** 99x
  - C** 101x
  - D** 100x

- 7** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** -1
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 1
  - D** 0
  - E** 101

- 8** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - E** Pole o 5 znacích a textový řetězec o délce 5 znaků

- 9** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** 9
  - C** 6
  - D** 5
  - E** 3

- 10** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 10 prvky s indexy 1..10 typu double
  - B** Pole s 9 prvky s indexy 0..9 typu double
  - C** Pole s 10 prvky s indexy 0..9 typu double
  - D** Pole s 9 prvky s indexy 1..9 typu double
  - E** kód se nepřeloží (obsahuje chybu)

**11** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 44
- B 66
- C 33
- D 0
- E 1

**12** Cyklus  
`int i = 99;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:

- A jednou
- B kód obsahuje syntaktickou chybu
- C dvakrát
- D ani jednou

**13** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 100
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 300
- E 200

**14** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World!"
- B "Hello World! , Hello World!!"
- C "!"
- D "Hello World! , Hello World!"
- E "Hello, Hello World!!"
- F kód se nepřeloží (obsahuje chybu)



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		73

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
```

bude obsah proměné A roven:

- A 97
- B 101
- C 100
- D 98
- E kód obsahuje syntaktickou chybu

**2** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 8 znaků
- B Pole o 7 znacích a textový řetězec o délce 7 znaků
- C Pole o 8 znacích a textový řetězec o délce 8 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 8 znacích a textový řetězec o délce 7 znaků

**3** Cyklus

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B 101x
- C 99x
- D 100x

**4** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A 6
- B 5
- C kód se nepřeloží (obsahuje chybu)
- D 9
- E 3

**5** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

- A 1
- B kód se nepřeloží (obsahuje chybu)
- C 0
- D -99
- E -100

**6** Datový typ int je:

- A Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- B Typ se znaménkem pro uchovávání celých čísel
- C Typ bez znaménka pro uchovávání celých čísel
- D Typ bez znaménka pro uchovávání čísel s desetinnou čárkou

**7** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
```

Bude obsah proměné A roven:

- A kód obsahuje syntaktickou chybu
- B 200
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 300
- E 100

**8** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu double
- B Pole s 9 prvky s indexy 0..9 typu double
- C Pole s 10 prvky s indexy 0..9 typu double
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 1..10 typu double

**9** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměné A roven:

- A 101
- B 300
- C 301
- D 200
- E 100
- F kód obsahuje syntaktickou chybu

- 10** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {`  
`}`  
proběhne:
- A** 0x
  - B** 99x
  - C** 100x
  - D** 101x
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 11** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:
- A** 4
  - B** 30
  - C** 60
  - D** 0
  - E** 40
- 
- 12** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** není definováno
  - B** nenulová hodnota
  - C** 0
  - D** 0 pokud je A rovno 0
- 
- 13** Cyklus  
`int i = 100;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** ani jednou
  - C** dvakrát
  - D** jednou
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```
- A** "Hello World!"
  - B** kód se nepřeloží (obsahuje chybu)
  - C** "!"
  - D** "Hello World! , Hello World!"
  - E** " , Hello World!!"
  - F** "Hello, Hello World!!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 74           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
```

proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 101x  
**D** 99x
- 2** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

**A** 100x  
**B** 101x  
**C** 99x  
**D** kód obsahuje syntaktickou chybu
- 3** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
```

Bude obsah proměnné A roven:

**A** 300  
**B** 100  
**C** kód obsahuje syntaktickou chybu  
**D** 200  
**E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 4** Hodnota výrazu  $(2-20)/(5+5)$  je:

**A** 1  
**B** 2  
**C** -1.8  
**D** -1  
**E** 0
- 5** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

**A** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 6 znaků
- 6** Datový typ int má vždy velikost:

**A** 4 bajty  
**B** 2 bajty  
**C** 8 bajty  
**D** v závislosti na architektuře
- 7** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

**A** 0  
**B** nenulová hodnota  
**C** není definováno  
**D** 0 pokud je A rovno 0
- 8** Následující zápis reprezentuje:

```
double x[10];
```

**A** Pole s 10 prvky s indexy 1..10 typu double  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** Pole s 10 prvky s indexy 0..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 9 prvky s indexy 1..9 typu double
- 9** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 6  
**C** 3  
**D** 5  
**E** 9
- 10** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x += 1; }
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 0  
**C** 100  
**D** 99  
**E** 1
- 11** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 100  
**C** 101  
**D** 102

- 12** Po provedení kódu:  
`int A = 100;`  
`A = A-- + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 299
  - C** 300
  - D** 100
  - E** 101
  - F** 200

- 13** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {`  
`}`  
proběhne:
- A** kód se nepřeloží (obsahuje chybu)
  - B** 100x
  - C** 0x
  - D** 101x
  - E** 99x

- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** "!"
  - C** "Hello World! , Hello World!"
  - D** " , Hello World!!"
  - E** "Hello World!"
  - F** "Hello, Hello World!!"

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		75

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

**A** 101x  
**B** 99x  
**C** kód obsahuje syntaktickou chybu  
**D** 100x
- 2** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 6  
**C** 5  
**D** 3  
**E** 9
- 3** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměnná typu int proběhne:

**A** vždy 100x  
**B** cyklus nemusí proběhnout ani jednou  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu
- 4** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "Hello World! , Hello World!"  
**B** "!"  
**C** "Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** " , Hello World!!"  
**F** "Hello, Hello World!!"
- 5** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

**A** -100  
**B** kód se nepřeloží (obsahuje chybu)  
**C** -99  
**D** 1  
**E** 0
- 6** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

**A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 7** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměnné A roven:

**A** 300  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 100  
**D** kód obsahuje syntaktickou chybu  
**E** 200
- 8** Hodnota výrazu  $(2+3)/5+5$  je:

**A** 10  
**B** 1  
**C** 6  
**D** 2  
**E** 0
- 9** Datový typ int má vždy velikost:

**A** 8 bajty  
**B** 2 bajty  
**C** 4 bajty  
**D** v závislosti na architektuře

- 10** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměné A roven:
- A** 102
  - B** 98
  - C** 100
  - D** 101
  - E** kód obsahuje syntaktickou chybu
- 
- 11** Následující zápis reprezentuje:  

```
int x[10];
```
- A** Pole s 10 prvků s indexy 1..10 typu int
  - B** Pole s 9 prvků s indexy 1..9 typu int
  - C** Pole s 10 prvků s indexy 0..9 typu int
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole s 9 prvků s indexy 0..9 typu int
- 
- 12** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměná typu integer je:
- A** 0
  - B** 0 pokud je A rovno 0
  - C** nenulová hodnota
  - D** není definováno
- 
- 13** Po provedení kódu:  

```
int A = 100;  
A = A-- + 200;
```

Bude obsah proměné A roven:
- A** 299
  - B** kód obsahuje syntaktickou chybu
  - C** 300
  - D** 101
  - E** 100
  - F** 200
- 
- 14** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
}
```

proběhne:
- A** kód se nepřeloží (obsahuje chybu)
  - B** 99x
  - C** 100x
  - D** 0x
  - E** 101x

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		76

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 5 znacích a textový řetězec o délce 6 znaků
- B** Pole o 5 znacích a textový řetězec o délce 5 znaků
- C** Pole o 6 znacích a textový řetězec o délce 6 znaků
- D** Pole o 6 znacích a textový řetězec o délce 5 znaků
- E** kód se nepřeloží (obsahuje chybu)

**2** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A** 0
- B** 30
- C** 60
- D** 40
- E** 4

**3** Datový typ char je:

- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- B** Typ pro uchovávání malých celých čísel nebo znaků
- C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D** Typ pro uchovávání velkých celých čísel

**4** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A** 101x
- B** 99x
- C** 100x
- D** kód obsahuje syntaktickou chybu

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** 9
- B** 6
- C** 3
- D** 5
- E** kód se nepřeloží (obsahuje chybu)

**6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

- A** 0
- B** kód se nepřeloží (obsahuje chybu)
- C** -100
- D** 1
- E** -99

**7** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 1..9 typu int
- B** Pole s 9 prvky s indexy 0..9 typu int
- C** Pole s 10 prvky s indexy 1..10 typu int
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole s 10 prvky s indexy 0..9 typu int

**8** Po provedení kódu:

```
int A = 100;
```

```
A = ++A + 200;
```

Bude obsah proměnné A roven:

- A** 301
- B** 300
- C** 100
- D** 200
- E** kód obsahuje syntaktickou chybu
- F** 101

**9** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:

- A** 10
- B** 0
- C** 1
- D** není definováno

**10** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A** " , Hello World!!"
- B** "Hello, Hello World!!"
- C** "!"
- D** kód se nepřeloží (obsahuje chybu)
- E** "Hello World! , Hello World!"
- F** "Hello World!"

**11** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:

- A** 100x
- B** 101x
- C** 99x
- D** 0x
- E** kód se nepřeloží (obsahuje chybu)

**12** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

Bude obsah proměnné A roven:

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** 300
- C** kód obsahuje syntaktickou chybu
- D** 200
- E** 100

**13** Cyklus

```
int i = 100;  
do {  
    i++;  
} while (i < 100);
```

proběhne:

- A** ani jednou
- B** kód obsahuje syntaktickou chybu
- C** jednou
- D** dvakrát

**14** Po provedení kódu:

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}
```

bude obsah proměnné A roven:

- A** 103
- B** 102
- C** kód obsahuje syntaktickou chybu
- D** 101



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		77

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 9 prvky s indexy 0..9 typu double
  - B** Pole s 9 prvky s indexy 1..9 typu double
  - C** Pole s 10 prvky s indexy 0..9 typu double
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole s 10 prvky s indexy 1..10 typu double

- 2** Cyklus  
`int i = 0;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:
- A** 99x
  - B** 100x
  - C** 101x
  - D** kód obsahuje syntaktickou chybu

- 3** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
`case 99: A++; break;`  
`case 100: A++; break;`  
`case 101: A++; break;`  
`default: A--;`  
`}`  
bude obsah proměnné A roven:
- A** 101
  - B** 100
  - C** 98
  - D** 97
  - E** kód obsahuje syntaktickou chybu

- 4** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** 1
  - B** 2
  - C** -1
  - D** 0
  - E** -1.8

- 5** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = **pA + 200;`  
Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** kód obsahuje syntaktickou chybu
  - C** 100
  - D** 300
  - E** 200

- 6** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
`i++;`  
`}`  
proběhne:
- A** 100x
  - B** kód obsahuje syntaktickou chybu
  - C** 99x
  - D** 101x

- 7** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i < 3; i++)`  
`y = y + x[i];`
- A** 9
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 3
  - D** 6
  - E** 5

- 8** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - B** Pole o 7 znacích a textový řetězec o délce 8 znaků
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - E** Pole o 8 znacích a textový řetězec o délce 7 znaků

- 9** Datový typ `int` má vždy velikost:
- A** v závislosti na architektuře
  - B** 2 bajty
  - C** 4 bajty
  - D** 8 bajty

- 10** Po vykonání následujícího kódu bude v `buf` uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcat(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** kód se nepřeloží (obsahuje chybu)
  - B** "!"
  - C** "Hello, Hello World!!"
  - D** "Hello World! , Hello World!"
  - E** "Hello World!"
  - F** "Hello World! , Hello World!!"

**11** Výsledkem výrazu (1 && 0) je:

- A 0
- B není definováno
- C 10
- D 1

**12** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x += 1; }
```

- A 99
- B kód se nepřeloží (obsahuje chybu)
- C 0
- D 100
- E 1

**13** Po provedení kódu:

```
int A = 100;  
A = A++ + 200;  
Bude obsah proměnné A roven:
```

- A 100
- B 200
- C 300
- D kód obsahuje syntaktickou chybu
- E 101
- F 301

**14** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {  
    }
```

proběhne:

- A 101x
- B 99x
- C 0x
- D kód se nepřeloží (obsahuje chybu)
- E 100x

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		78

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Datový typ int je:
- A** Typ se znaménkem pro uchovávání celých čísel
  - B** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
  - C** Typ bez znaménka pro uchovávání celých čísel
  - D** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
- 2** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 66
  - B** 44
  - C** 0
  - D** 33
  - E** 1
- 3** Po provedení kódu:
- ```
int A = 100;  
A = A-- + 200;
```
- Bude obsah proměnné A roven:
- A** 100
  - B** 200
  - C** 300
  - D** 101
  - E** kód obsahuje syntaktickou chybu
  - F** 299
- 4** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:
- A** 0 pokud je A rovno 0
  - B** 1
  - C** 0
  - D** není definováno
- 5** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x -= 1; }
```
- A** 1
  - B** kód se nepřeloží (obsahuje chybu)
  - C** -99
  - D** 0
  - E** -100
- 6** Cyklus
- ```
int i = 99;  
do {  
    i++;  
} while (i < 100);
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** dvakrát
  - C** jednou
  - D** ani jednou
- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```
- A** " , Hello World!!"
  - B** "Hello World! , Hello World!"
  - C** kód se nepřeloží (obsahuje chybu)
  - D** "!"
  - E** "Hello World!"
  - F** "Hello, Hello World!!"
- 8** Cyklus
- ```
int i = 0;  
while (i < 100) {  
    i++;  
}
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** 101x
  - C** 100x
  - D** 99x
- 9** Tělo cyklu
- ```
for (int i = 0; i <= 100; i++) {  
}
```
- proběhne:
- A** 0x
  - B** 99x
  - C** 100x
  - D** 101x
  - E** kód se nepřeloží (obsahuje chybu)
- 10** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - E** kód se nepřeloží (obsahuje chybu)

- 11** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}
```

bude obsah proměnné A roven:
- A** 101
  - B** 102
  - C** 103
  - D** kód obsahuje syntaktickou chybu
- 
- 12** Následující zápis reprezentuje:  

```
double x[10];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole s 10 prvků s indexy 0..9 typu double
  - C** Pole s 9 prvků s indexy 1..9 typu double
  - D** Pole s 9 prvků s indexy 0..9 typu double
  - E** Pole s 10 prvků s indexy 1..10 typu double
- 
- 13** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```
- A** 6
  - B** 9
  - C** 5
  - D** 3
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 14** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;
```

Bude obsah proměnné A roven:
- A** 200
  - B** kód obsahuje syntaktickou chybu
  - C** 300
  - D** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - E** 100

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 79           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole o 7 znacích a textový řetězec o délce 7 znaků
- C** Pole o 8 znacích a textový řetězec o délce 8 znaků
- D** Pole o 7 znacích a textový řetězec o délce 8 znaků
- E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**2** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A** 60
- B** 40
- C** 4
- D** 0
- E** 30

**3** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:

- A** není definováno
- B** 0 pokud je A rovno 0
- C** 1
- D** 0

**4** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
```

bude obsah proměné A roven:

- A** 97
- B** 100
- C** 98
- D** 101
- E** kód obsahuje syntaktickou chybu

**5** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {`  
`}`  
proběhne:

- A** 0x
- B** kód se nepřeloží (obsahuje chybu)
- C** 101x
- D** 99x
- E** 100x

**6** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměné A roven:

- A** 300
- B** 100
- C** 101
- D** kód obsahuje syntaktickou chybu
- E** 200
- F** 299

**7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "!"
- B** "Hello World!"
- C** "Hello, Hello World!!"
- D** "Hello World! , Hello World!"
- E** kód se nepřeloží (obsahuje chybu)
- F** "Hello World! , Hello World!!"

**8** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměná typu int proběhne:

- A** alespoň jednou
- B** kód obsahuje syntaktickou chybu
- C** vždy 100x
- D** cyklus nemusí proběhnout ani jednou

**9** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++) {
    x += 1; }
```

- A** 1
- B** 100
- C** 0
- D** 99
- E** kód se nepřeloží (obsahuje chybu)

- 10** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** alespoň jednou  
**B** kód obsahuje syntaktickou chybu  
**C** cyklus nemusí proběhnout ani jednou  
**D** vždy 100x
- 
- 11** Datový typ char je:  
**A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**C** Typ pro uchovávání malých celých čísel nebo znaků  
**D** Typ pro uchovávání velkých celých čísel
- 
- 12** Hodnota proměnné po provedení následujícího kódu bude:  
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 3  
**C** 9  
**D** 5  
**E** 6
- 
- 13** Následující zápis reprezentuje:  
int x[10];  
**A** Pole s 9 prvky s indexy 0..9 typu int  
**B** Pole s 10 prvky s indexy 1..10 typu int  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 10 prvky s indexy 0..9 typu int  
**E** Pole s 9 prvky s indexy 1..9 typu int
- 
- 14** Po provedení kódu:  
int A = 100;  
int\* pA = &A;  
A = \*pA + 200;  
Bude obsah proměnné A roven:  
**A** 200  
**B** 100  
**C** 300  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 80           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Cyklus  

```
int i = 99;  
do {  
    i++;  
} while (i < 100);
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B dvakrát
- C jednou
- D ani jednou

**2** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
}
```

proběhne:

- A 100x
- B 101x
- C 0x
- D 99x
- E kód se nepřeloží (obsahuje chybu)

**3** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B 101x
- C 100x
- D 99x

**4** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x += 1; }
```

- A 99
- B kód se nepřeloží (obsahuje chybu)
- C 0
- D 1
- E 100

**5** Datový typ int má vždy velikost:

- A 4 bajty
- B 2 bajty
- C v závislosti na architektuře
- D 8 bajty

**6** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World! , Hello World!"
- B "Hello World!"
- C "Hello World! , Hello World!"
- D "Hello, Hello World!"
- E kód se nepřeloží (obsahuje chybu)
- F "!"

**7** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 9
- C 3
- D 5
- E 6

**8** Výsledkem výrazu (1 && 0) je:

- A není definováno
- B 0
- C 1
- D 10

**9** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;
```

Bude obsah proměně A roven:

- A 100
- B 200
- C kód obsahuje syntaktickou chybu
- D obsah hodnoty A nelze bez spuštění jednoznačně určit
- E 300

**10** Po provedení kódu:  

```
int A = 100;  
A = A++ + 200;
```

Bude obsah proměně A roven:

- A 101
- B 100
- C 200
- D kód obsahuje syntaktickou chybu
- E 301
- F 300

**11** Hodnota výrazu  $(2+3)/5+5$  je:

- A** 1
- B** 0
- C** 2
- D** 6
- E** 10

**12** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
- B** Pole o 7 znacích a textový řetězec o délce 7 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 8 znacích a textový řetězec o délce 7 znaků
- E** Pole o 7 znacích a textový řetězec o délce 8 znaků

**13** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 100
- C** 102
- D** 101

**14** Následující zápis reprezentuje:

```
int x[10];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole s 10 prvky s indexy 0..9 typu int
- C** Pole s 9 prvky s indexy 0..9 typu int
- D** Pole s 9 prvky s indexy 1..9 typu int
- E** Pole s 10 prvky s indexy 1..10 typu int



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 81           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměnné A roven:

**A** 100  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** kód obsahuje syntaktickou chybu  
**D** 300  
**E** 200
- 2** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

**A** nenulová hodnota  
**B** není definováno  
**C** 0  
**D** 0 pokud je A rovno 0
- 3** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
y = y + x[i];`

**A** 9  
**B** 3  
**C** 6  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 5
- 4** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }`

**A** kód se nepřeloží (obsahuje chybu)  
**B** 1  
**C** -100  
**D** 0  
**E** -99
- 5** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`

**A** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**C** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 6** Cyklus  
`while (i < 100) {  
i++;  
}`  
kde i je proměnná typu int proběhne:

**A** kód obsahuje syntaktickou chybu  
**B** cyklus nemusí proběhnout ani jednou  
**C** vždy 100x  
**D** alespoň jednou
- 7** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {  
}`  
proběhne:

**A** 99x  
**B** 101x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 100x  
**E** 0x
- 8** Následující zápis reprezentuje:  
`double x[10];`

**A** Pole s 9 prvky s indexy 0..9 typu double  
**B** Pole s 10 prvky s indexy 1..10 typu double  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** Pole s 9 prvky s indexy 1..9 typu double
- 9** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");`

**A** "Hello World! , Hello World!"  
**B** "!"  
**C** "Hello World!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello World! , Hello World!!"  
**F** "Hello, Hello World!!"
- 10** Po provedení kódu:  
`int A = 100;  
A = A++ + 200;`  
Bude obsah proměnné A roven:

**A** 300  
**B** 301  
**C** 100  
**D** kód obsahuje syntaktickou chybu  
**E** 200  
**F** 101

**11** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 102
- C** 103
- D** 101

**12** Datový typ float je:

- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- B** Typ se znaménkem pro uchovávání celých čísel
- C** Typ bez znaménka pro uchovávání celých čísel
- D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**13** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A** 0
- B** 33
- C** 1
- D** 44
- E** 66

**14** Cyklus

```
int i = 100;
do {
    i++;
} while (i < 100);
```

proběhne:

- A** kód obsahuje syntaktickou chybu
- B** jednou
- C** dvakrát
- D** ani jednou

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 82           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {  
}  
proběhne:`  
**A** 0x  
**B** 101x  
**C** 100x  
**D** 99x  
**E** kód se nepřeloží (obsahuje chybu)
- 2** Následující zápis reprezentuje:  
`double x[10];`  
**A** Pole s 9 prvky s indexy 0..9 typu double  
**B** Pole s 9 prvky s indexy 1..9 typu double  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** Pole s 10 prvky s indexy 1..10 typu double
- 3** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 6  
**B** 10  
**C** 0  
**D** 1  
**E** 2
- 4** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++){  
{ x += 1; }`  
**A** 1  
**B** 0  
**C** 99  
**D** 100  
**E** kód se nepřeloží (obsahuje chybu)
- 5** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** "Hello World! , Hello World!"  
**C** "Hello, Hello World!!"  
**D** " , Hello World!!"  
**E** "Hello World!"  
**F** "!"
- 6** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:  
**A** nenulová hodnota  
**B** 0 pokud je A rovno 0  
**C** 0  
**D** není definováno
- 7** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++){  
y = y + x[i];`  
**A** 5  
**B** 9  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 6  
**E** 3
- 8** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměnné A roven:  
**A** 100  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 200  
**D** kód obsahuje syntaktickou chybu  
**E** 300
- 9** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků
- 10** Cyklus  
`int i = 0;  
while (i <= 100) {  
i++;  
}  
proběhne:`  
**A** kód obsahuje syntaktickou chybu  
**B** 99x  
**C** 101x  
**D** 100x

- 11** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 301
  - C** 100
  - D** 200
  - E** kód obsahuje syntaktickou chybu
  - F** 300

- 12** Cyklus  
`int i = 100;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** ani jednou
  - C** jednou
  - D** dvakrát

- 13** Datový typ float je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - C** Typ se znaménkem pro uchovávání celých čísel
  - D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

- 14** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
    `case 100: A++;`  
    `case 101: A++; break;`  
    `case 102: A++;`  
`}`  
bude obsah proměnné A roven:
- A** 101
  - B** 102
  - C** 103
  - D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 83           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Datový typ int má vždy velikost:

- A 2 bajty
- B 8 bajty
- C v závislosti na architektuře
- D 4 bajty

**2** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "!"
- C "Hello World! , Hello World!"
- D "Hello, Hello World!!"
- E "Hello World!"
- F "Hello World! , Hello World!!"

**3** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměně A roven:

- A 101
- B kód obsahuje syntaktickou chybu
- C 102
- D 100

**4** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:

- A 0x
- B 99x
- C 100x
- D 101x
- E kód se nepřeloží (obsahuje chybu)

**5** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměná typu int proběhne:

- A vždy 100x
- B kód obsahuje syntaktickou chybu
- C cyklus nemusí proběhnout ani jednou
- D alespoň jednou

**6** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměně A roven:

- A 100
- B 200
- C 300
- D 101
- E kód obsahuje syntaktickou chybu
- F 299

**7** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++) {
    x += 1;
}
```

- A 0
- B 1
- C kód se nepřeloží (obsahuje chybu)
- D 100
- E 99

**8** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměná typu integer je:

- A není definováno
- B 0
- C nenulová hodnota
- D 0 pokud je A rovno 0

**9** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměná typu int proběhne:

- A vždy 100x
- B alespoň jednou
- C kód obsahuje syntaktickou chybu
- D cyklus nemusí proběhnout ani jednou

**10** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 7 znaků
- B Pole o 8 znacích a textový řetězec o délce 7 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 8 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 8 znaků

**11** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 2
- B 0
- C -1
- D 1
- E -1.8

**12** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 9
- C** 3
- D** 6
- E** 5

**13** Následující zápis reprezentuje:

```
double x[10];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole s 10 prvků s indexy 1..10 typu double
- C** Pole s 10 prvků s indexy 0..9 typu double
- D** Pole s 9 prvků s indexy 1..9 typu double
- E** Pole s 9 prvků s indexy 0..9 typu double

**14** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = pA + 200;  
Bude obsah proměnné A roven:
```

- A** 200
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 300
- E** obsah hodnoty A nelze bez spuštění jednoznačně určit

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 84           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 60
- B 0
- C 40
- D 4
- E 30

**2** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:

- A 0 pokud je A rovno 0
- B 1
- C není definováno
- D 0

**3** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *pA + 200;
Bude obsah proměné A roven:
```

- A 100
- B 300
- C kód obsahuje syntaktickou chybu
- D 200
- E obsah hodnoty A nelze bez spuštění jednoznačně určit

**4** Datový typ int je:

- A Typ se znaménkem pro uchovávání celých čísel
- B Typ bez znaménka pro uchovávání celých čísel
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ se znaménkem pro uchovávání čísel s desetinou čárkou

**5** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++) {
    x += 1;
}
```

- A 1
- B 0
- C 99
- D 100
- E kód se nepřeloží (obsahuje chybu)

**6** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 8 znaků
- B Pole o 8 znacích a textový řetězec o délce 7 znaků
- C Pole o 8 znacích a textový řetězec o délce 8 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 7 znacích a textový řetězec o délce 7 znaků

**7** Následující zápis reprezentuje:  
`double x[10];`

- A Pole s 9 prvky s indexy 1..9 typu double
- B Pole s 9 prvky s indexy 0..9 typu double
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 10 prvky s indexy 1..10 typu double
- E Pole s 10 prvky s indexy 0..9 typu double

**8** Cyklus  

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A 99x
- B kód obsahuje syntaktickou chybu
- C 100x
- D 101x

**9** Po provedení kódu:  

```
int A = 100;
A = A++ + 200;
```

Bude obsah proměné A roven:

- A kód obsahuje syntaktickou chybu
- B 101
- C 100
- D 301
- E 200
- F 300

**10** Cyklus  

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

- A dvakrát
- B kód obsahuje syntaktickou chybu
- C jednou
- D ani jednou

**11** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {
    }
```

proběhne:

- A 99x
- B 101x
- C 0x
- D kód se nepřeloží (obsahuje chybu)
- E 100x

- 12** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 5  
**B** 6  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 3  
**E** 9
- 
- 13** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** 101  
**B** kód obsahuje syntaktickou chybu  
**C** 98  
**D** 100  
**E** 102
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```
- A** " , Hello World!!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** "Hello World!"  
**D** "Hello World! , Hello World!"  
**E** "Hello, Hello World!!"  
**F** "!"



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		85

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");`
- A** "Hello World! , Hello World!"  
**B** "!"  
**C** "Hello World!"  
**D** "Hello, Hello World!!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** " , Hello World!!"

- 2** Po provedení kódu:  
`int A = 98;  
switch (A) {  
 case 99: A++; break;  
 case 100: A++; break;  
 case 101: A++; break;  
}`  
bude obsah proměné A roven:
- A** 101  
**B** 98  
**C** kód obsahuje syntaktickou chybu  
**D** 102  
**E** 100

- 3** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** 2  
**B** -1.8  
**C** 0  
**D** -1  
**E** 1

- 4** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměné A roven:
- A** 200  
**B** kód obsahuje syntaktickou chybu  
**C** 100  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 300

- 5** Tělo cyklu  
`for (int i = 1; i < 100; i++) {  
}`  
proběhne:
- A** 101x  
**B** 99x  
**C** 100x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 0x

- 6** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:
- A** 10  
**B** 0  
**C** 1  
**D** není definováno

- 7** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
 y = y + x[i];`
- A** 6  
**B** 5  
**C** 3  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 9

- 8** Po provedení kódu:  
`int A = 100;  
A = A-- + 200;`  
Bude obsah proměné A roven:
- A** 300  
**B** 200  
**C** 100  
**D** kód obsahuje syntaktickou chybu  
**E** 299  
**F** 101

- 9** Cyklus  
`int i = 0;  
while (i <= 100) {  
 i++;  
}`  
proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** 101x  
**C** 99x  
**D** 100x

**10** Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 9 prvky s indexy 0..9 typu int
- B Pole s 9 prvky s indexy 1..9 typu int
- C Pole s 10 prvky s indexy 1..10 typu int
- D Pole s 10 prvky s indexy 0..9 typu int
- E kód se nepřeloží (obsahuje chybu)

**11** Datový typ char je:

- A Typ pro uchovávání malých celých čísel nebo znaků
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ pro uchovávání velkých celých čísel

**12** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 8 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 7 znacích a textový řetězec o délce 8 znaků
- D Pole o 7 znacích a textový řetězec o délce 7 znaků
- E Pole o 8 znacích a textový řetězec o délce 7 znaků

**13** Cyklus

```
int i = 100;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A dvakrát
- B kód obsahuje syntaktickou chybu
- C jednou
- D ani jednou

**14** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A 101
- B -1
- C 0
- D 1
- E kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		86

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 6  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 5  
**D** 3  
**E** 9
- 2** Následující zápis reprezentuje:

```
int x[10];
```

**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** Pole s 9 prvky s indexy 0..9 typu int  
**C** Pole s 9 prvky s indexy 1..9 typu int  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 0..9 typu int
- 3** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:

**A** kód se nepřeloží (obsahuje chybu)  
**B** 101x  
**C** 100x  
**D** 99x  
**E** 0x
- 4** Hodnota výrazu  $(2-20)/(5+5)$  je:

**A** -1.8  
**B** 1  
**C** 2  
**D** -1  
**E** 0
- 5** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměnná typu int proběhne:

**A** cyklus nemusí proběhnout ani jednou  
**B** vždy 100x  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu
- 6** Datový typ int je:

**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou  
**C** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel
- 7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "Hello World!"  
**B** "!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** " , Hello World!"  
**E** "Hello, Hello World!!"  
**F** "Hello World! , Hello World!"
- 8** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměnné A roven:

**A** 100  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 300  
**D** kód obsahuje syntaktickou chybu  
**E** 200
- 9** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 103  
**C** 101  
**D** 102
- 10** Výsledkem výrazu  $(1 \&\& 1)$  je:

**A** 0  
**B** 10  
**C** není definováno  
**D** 1

- 11** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** 0  
**B** 101  
**C** -1  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 1

- 12** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```
- A** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 13** Po provedení kódu:
- ```
int A = 100;
A = --A + 200;
```
- Bude obsah proměnné A roven:
- A** 200  
**B** 299  
**C** kód obsahuje syntaktickou chybu  
**D** 101  
**E** 300  
**F** 100

- 14** Cyklus
- ```
int i = 0;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** 101x  
**B** 100x  
**C** 99x  
**D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		87

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující zápis reprezentuje:  
`double x[10];`

- A** Pole s 10 prvky s indexy 0..9 typu double
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 9 prvky s indexy 0..9 typu double
- D** Pole s 10 prvky s indexy 1..10 typu double
- E** Pole s 9 prvky s indexy 1..9 typu double

**2** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:

- A** 301
- B** 101
- C** kód obsahuje syntaktickou chybu
- D** 100
- E** 200
- F** 300

**3** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {`  
`}`  
proběhne:

- A** 0x
- B** 99x
- C** 101x
- D** 100x
- E** kód se nepřeloží (obsahuje chybu)

**4** Výsledkem výrazu `(1 && 0)` je:

- A** 1
- B** 0
- C** 10
- D** není definováno

**5** Cyklus  
`int i = 100;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:

- A** ani jednou
- B** dvakrát
- C** jednou
- D** kód obsahuje syntaktickou chybu

**6** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 6
- C** 3
- D** 5
- E** 9

**7** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`

- A** Pole o 6 znacích a textový řetězec o délce 5 znaků
- B** Pole o 5 znacích a textový řetězec o délce 6 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 5 znacích a textový řetězec o délce 5 znaků
- E** Pole o 6 znacích a textový řetězec o délce 6 znaků

**8** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
`i++;`  
`}`  
proběhne:

- A** kód obsahuje syntaktickou chybu
- B** 99x
- C** 100x
- D** 101x

**9** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
`case 99: A++; break;`  
`case 100: A++; break;`  
`case 101: A++; break;`  
`}`  
bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 100
- C** 102
- D** 101

- 10** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");`
- A** "!"  
**B** "Hello, Hello World!"  
**C** "Hello World! , Hello World!"  
**D** "Hello World!"  
**E** " , Hello World!"  
**F** kód se nepřeloží (obsahuje chybu)
- 
- 11** Datový typ `int` má vždy velikost:
- A** v závislosti na architektuře  
**B** 4 bajty  
**C** 8 bajty  
**D** 2 bajty
- 
- 12** Hodnota proměnné `x` po provedení následující kódu bude:  
`int x = 0;  
if (1)  
 x = x + 1;  
else  
 x = x - 1;  
for (int i = 0; i < 100; i++) {x++};`
- A** 0  
**B** 1  
**C** 101  
**D** -1  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:
- A** 40  
**B** 30  
**C** 4  
**D** 0  
**E** 60
- 
- 14** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměnné `A` roven:
- A** 200  
**B** 100  
**C** obsah hodnoty `A` nelze bez spuštění jednoznačně určit  
**D** 300  
**E** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		88

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

**A** "Hello, Hello World!!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** "Hello World!"  
**D** "!"  
**E** "Hello World! , Hello World!"  
**F** "Hello World! , Hello World!!"

- 2** Cyklus  

```
do {  
    i++;  
} while (i < 100);
```

kde i je proměná typu int proběhne:  
**A** alespoň jednou  
**B** vždy 100x  
**C** kód obsahuje syntaktickou chybu  
**D** cyklus nemusí proběhnout ani jednou

- 3** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** 101x  
**C** 99x  
**D** 100x

- 4** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x += 1; }
```

**A** 1  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 100  
**D** 0  
**E** 99

- 5** Hodnota výrazu (2+3)/5+5 je:  
**A** 1  
**B** 6  
**C** 10  
**D** 2  
**E** 0

- 6** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:  
**A** 100  
**B** 102  
**C** 101  
**D** kód obsahuje syntaktickou chybu

- 7** Tělo cyklu  

```
for (int i = 0; i <= 100; i++) {  
}
```

proběhne:  
**A** 99x  
**B** 0x  
**C** 100x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101x

- 8** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

**A** 9  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 6  
**D** 5  
**E** 3

- 9** Výsledkem výrazu (0 || A), kde A je proměnná typu integer je:  
**A** 0 pokud je A rovno 0  
**B** 1  
**C** 0  
**D** není definováno

- 10** Datový typ int je:  
**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou  
**C** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel

- 11** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 10 prvky s indexy 0..9 typu double
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole s 10 prvky s indexy 1..10 typu double
  - D** Pole s 9 prvky s indexy 1..9 typu double
  - E** Pole s 9 prvky s indexy 0..9 typu double
- 
- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *pA + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - C** 100
  - D** 200
  - E** 300
- 
- 13** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - C** Pole o 7 znacích a textový řetězec o délce 8 znaků
  - D** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - E** Pole o 8 znacích a textový řetězec o délce 7 znaků
- 
- 14** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** kód obsahuje syntaktickou chybu
  - C** 301
  - D** 200
  - E** 100
  - F** 101



# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		89

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 9
- B kód se nepřeloží (obsahuje chybu)
- C 3
- D 5
- E 6

**2** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

Bude obsah proměnné A roven:  

- A kód obsahuje syntaktickou chybu
- B 100
- C 200
- D 300
- E obsah hodnoty A nelze bez spuštění jednoznačně určit

**3** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
}  
proběhne:
```

- A 101x
- B kód se nepřeloží (obsahuje chybu)
- C 0x
- D 99x
- E 100x

**4** Po provedení kódu:  

```
int A = 100;  
A = --A + 200;
```

Bude obsah proměnné A roven:  

- A 101
- B 299
- C 300
- D kód obsahuje syntaktickou chybu
- E 200
- F 100

**5** Následující zápis reprezentuje:  

```
int x[10];
```

- A Pole s 10 prvky s indexy 0..9 typu int
- B Pole s 9 prvky s indexy 0..9 typu int
- C Pole s 9 prvky s indexy 1..9 typu int
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 1..10 typu int

**6** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:  

- A 10
- B 0
- C 1
- D není definováno

**7** Cyklus  

```
int i = 0;  
do {  
    i++;  
} while (i < 100);
```

proběhne:  

- A 99x
- B 100x
- C 101x
- D kód obsahuje syntaktickou chybu

**8** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "Hello World! , Hello World!"
- C "Hello World!"
- D " , Hello World!!"
- E "!"
- F "Hello, Hello World!!"

**9** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:  

- A 40
- B 60
- C 4
- D 0
- E 30

**10** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

- A Pole o 6 znacích a textový řetězec o délce 6 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 6 znacích a textový řetězec o délce 5 znaků
- D Pole o 5 znacích a textový řetězec o délce 5 znaků
- E Pole o 5 znacích a textový řetězec o délce 6 znaků

- 11** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}
```

bude obsah proměnné A roven:
- A** 100
  - B** 97
  - C** 98
  - D** 101
  - E** kód obsahuje syntaktickou chybu
- 
- 12** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x += 1; }
```
- A** 1
  - B** 99
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 100
  - E** 0
- 
- 13** Datový typ int má vždy velikost:
- A** 4 bajty
  - B** 8 bajty
  - C** v závislosti na architektuře
  - D** 2 bajty
- 
- 14** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}
```

proběhne:
- A** 101x
  - B** 99x
  - C** 100x
  - D** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		90

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 40
- B 60
- C 0
- D 4
- E 30

**2** Tělo cyklu  
`for (int i = 0; i < 100; i++) {`  
`}`  
proběhne:

- A 99x
- B 100x
- C 0x
- D 101x
- E kód se nepřeloží (obsahuje chybu)

**3** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`

- A Pole o 5 znacích a textový řetězec o délce 6 znaků
- B Pole o 6 znacích a textový řetězec o délce 5 znaků
- C Pole o 5 znacích a textový řetězec o délce 5 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

**4** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:

- A 10
- B není definováno
- C 0
- D 1

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 6
- B 3
- C kód se nepřeloží (obsahuje chybu)
- D 9
- E 5

**6** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A 102
- B 100
- C 98
- D kód obsahuje syntaktickou chybu
- E 101

**7** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

- A 1
- B -100
- C -99
- D 0
- E kód se nepřeloží (obsahuje chybu)

**8** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A "!"
- B "Hello, Hello World!!"
- C kód se nepřeloží (obsahuje chybu)
- D "Hello World! , Hello World!!"
- E "Hello World!"
- F "Hello World! , Hello World!"

**9** Datový typ char je:

- A Typ pro uchovávání malých celých čísel nebo znaků
- B Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
- C Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
- D Typ pro uchovávání velkých celých čísel

**10** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu double
- B kód se nepřeloží (obsahuje chybu)
- C Pole s 10 prvky s indexy 0..9 typu double
- D Pole s 9 prvky s indexy 0..9 typu double
- E Pole s 10 prvky s indexy 1..10 typu double

- 11** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** vždy 100x  
**B** alespoň jednou  
**C** kód obsahuje syntaktickou chybu  
**D** cyklus nemusí proběhnout ani jednou

- 12** Po provedení kódu:  
int A = 100;  
int\* pA = &A;  
A = \*A + 200;  
Bude obsah proměné A roven:  
**A** 200  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 100

- 13** Po provedení kódu:  
int A = 100;  
A = --A + 200;  
Bude obsah proměné A roven:  
**A** 300  
**B** kód obsahuje syntaktickou chybu  
**C** 101  
**D** 100  
**E** 299  
**F** 200

- 14** Cyklus  
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:  
**A** 100x  
**B** kód obsahuje syntaktickou chybu  
**C** 101x  
**D** 99x

# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		91

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 0
- B -1
- C 2
- D 1
- E -1.8

**2** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 3
- B 6
- C 9
- D kód se nepřeloží (obsahuje chybu)
- E 5

**3** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
    `i++;`  
`}`  
proběhne:

- A 99x
- B 100x
- C 101x
- D kód obsahuje syntaktickou chybu

**4** Datový typ float je:

- A Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- B Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- C Typ se znaménkem pro uchovávání celých čísel
- D Typ bez znaménka pro uchovávání celých čísel

**5** Po provedení kódu:

```
int A = 100;
A = A++ + 200;
Bude obsah proměnné A roven:
```

- A 100
- B 300
- C 200
- D 101
- E 301
- F kód obsahuje syntaktickou chybu

**6** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A 10
- B 0
- C není definováno
- D 1

**7** Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu int
- B Pole s 9 prvky s indexy 0..9 typu int
- C Pole s 10 prvky s indexy 0..9 typu int
- D Pole s 10 prvky s indexy 1..10 typu int
- E kód se nepřeloží (obsahuje chybu)

**8** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "Hello World! , Hello World!!"
- C "Hello World!"
- D "Hello, Hello World!!"
- E "Hello World! , Hello World!"
- F "!"

**9** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {`  
`}`  
proběhne:

- A 99x
- B 0x
- C 101x
- D 100x
- E kód se nepřeloží (obsahuje chybu)

**10** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A 0
- B 1
- C kód se nepřeloží (obsahuje chybu)
- D -1
- E 101

**11** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 5 znacích a textový řetězec o délce 6 znaků
- B Pole o 5 znacích a textový řetězec o délce 5 znaků
- C Pole o 6 znacích a textový řetězec o délce 5 znaků
- D kód se nepřeloží (obsahuje chybu)
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

- 12** Cyklus
- ```
int i = 99;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** dvakrát
  - C** jednou
  - D** ani jednou

- 13** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = **pA + 200;
```
- Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 200
  - C** 100
  - D** 300
  - E** obsah hodnoty A nelze bez spuštění jednoznačně určit

- 14** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```
- bude obsah proměnné A roven:
- A** 103
  - B** kód obsahuje syntaktickou chybu
  - C** 102
  - D** 101

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 92           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující zápis reprezentuje:  
`double x[10];`

- A** Pole s 10 prvky s indexy 1..10 typu double
- B** Pole s 10 prvky s indexy 0..9 typu double
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole s 9 prvky s indexy 1..9 typu double
- E** Pole s 9 prvky s indexy 0..9 typu double

**2** Hodnota výrazu  $(2+3)/5+5$  je:

- A** 1
- B** 0
- C** 2
- D** 10
- E** 6

**3** Hodnota proměnné `x` po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** -1
- B** kód se nepřeloží (obsahuje chybu)
- C** 0
- D** 101
- E** 1

**4** Po provedení kódu:

```
int A = 100;
A = ++A + 200;
Bude obsah proměně A roven:
```

- A** 300
- B** 100
- C** 301
- D** kód obsahuje syntaktickou chybu
- E** 101
- F** 200

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** 5
- B** 6
- C** kód se nepřeloží (obsahuje chybu)
- D** 9
- E** 3

**6** Po vykonání následujícího kódu bude v `buf` uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello World! , Hello World!!"
- B** kód se nepřeloží (obsahuje chybu)
- C** "Hello World!"
- D** "!"
- E** "Hello, Hello World!!"
- F** "Hello World! , Hello World!"

**7** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměně `A` roven:

- A** 102
- B** kód obsahuje syntaktickou chybu
- C** 101
- D** 103

**8** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A** 1
- B** 10
- C** není definováno
- D** 0

**9** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:

- A** kód se nepřeloží (obsahuje chybu)
- B** 0x
- C** 99x
- D** 101x
- E** 100x

**10** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
```

proběhne:

- A** 99x
- B** 100x
- C** 101x
- D** kód obsahuje syntaktickou chybu

- 11** Datový typ float je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - D** Typ se znaménkem pro uchovávání celých čísel

- 12** Po provedení kódu:
- ```
int A = 100;  
int* pA = &A;  
A = &pA + 200;
```
- Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** kód obsahuje syntaktickou chybu
  - C** 200
  - D** 100
  - E** 300

- 13** Cyklus
- ```
while (i < 100) {  
    i++;  
}
```
- kde i je proměnná typu int proběhne:
- A** cyklus nemusí proběhnout ani jednou
  - B** kód obsahuje syntaktickou chybu
  - C** alespoň jednou
  - D** vždy 100x

- 14** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - C** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - E** kód se nepřeloží (obsahuje chybu)



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		93

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
    }  
proběhne:
```

  
**A** 101x  
**B** 0x  
**C** 99x  
**D** 100x  
**E** kód se nepřeloží (obsahuje chybu)

- 2** Datový typ int je:  
**A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání celých čísel  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

- 3** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}  
bude obsah proměné A roven:
```

  
**A** 103  
**B** 101  
**C** 102  
**D** kód obsahuje syntaktickou chybu

- 4** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}  
proběhne:
```

  
**A** 100x  
**B** kód obsahuje syntaktickou chybu  
**C** 99x  
**D** 101x

- 5** Po provedení kódu:  

```
int A = 100;  
A = A++ + 200;  
Bude obsah proměné A roven:
```

  
**A** 100  
**B** 300  
**C** 301  
**D** 200  
**E** kód obsahuje syntaktickou chybu  
**F** 101

- 6** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

  
**A** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**E** Pole o 8 znacích a textový řetězec o délce 8 znaků

- 7** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *pA + 200;  
Bude obsah proměné A roven:
```

  
**A** 200  
**B** kód obsahuje syntaktickou chybu  
**C** 300  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 100

- 8** Následující zápis reprezentuje:  

```
int x[10];
```

  
**A** Pole s 9 prvky s indexy 0..9 typu int  
**B** Pole s 10 prvky s indexy 0..9 typu int  
**C** Pole s 9 prvky s indexy 1..9 typu int  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 1..10 typu int

- 9** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 33  
**B** 66  
**C** 0  
**D** 1  
**E** 44

- 10** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

  
**A** 101  
**B** 0  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 1  
**E** -1

- 11** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** není definováno
  - B** 0
  - C** 0 pokud je A rovno 0
  - D** nenulová hodnota

- 12** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World! , Hello World!"
  - B** "!"
  - C** "Hello World!"
  - D** kód se nepřeloží (obsahuje chybu)
  - E** "Hello World! , Hello World!!"
  - F** "Hello, Hello World!!"

- 13** Cyklus
- ```
int i = 100;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** ani jednou
  - B** jednou
  - C** kód obsahuje syntaktickou chybu
  - D** dvakrát

- 14** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** 6
  - C** 3
  - D** 9
  - E** 5

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 94           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`  
**A** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**B** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 2** Cyklus  
`int i = 100;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:  
**A** jednou  
**B** ani jednou  
**C** kód obsahuje syntaktickou chybu  
**D** dvakrát
- 3** Datový typ `int` má vždy velikost:  
**A** 2 bajty  
**B** v závislosti na architektuře  
**C** 8 bajty  
**D** 4 bajty
- 4** Výsledkem výrazu `(1 && 0)` je:  
**A** 0  
**B** 10  
**C** 1  
**D** není definováno
- 5** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
`i++;`  
`}`  
proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** 101x  
**C** 100x  
**D** 99x
- 6** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {`  
`}`  
proběhne:  
**A** 99x  
**B** 100x  
**C** 0x  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101x
- 7** Hodnota proměnné `x` po provedení následující kódu bude:  
`int x = 0;`  
`if (1)`  
`x = x + 1;`  
`else`  
`x = x - 1;`  
`for (int i = 0; i < 100; i++) {x++};`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 1  
**C** -1  
**D** 0  
**E** 101
- 8** Následující zápis reprezentuje:  
`double x[10];`  
**A** Pole s 10 prvky s indexy 1..10 typu `double`  
**B** Pole s 9 prvky s indexy 0..9 typu `double`  
**C** Pole s 10 prvky s indexy 0..9 typu `double`  
**D** Pole s 9 prvky s indexy 1..9 typu `double`  
**E** kód se nepřeloží (obsahuje chybu)
- 9** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:  
**A** 0  
**B** 40  
**C** 4  
**D** 60  
**E** 30
- 10** Po provedení kódu:  
`int A = 100;`  
`A = --A + 200;`  
Bude obsah proměnné `A` roven:  
**A** 300  
**B** 101  
**C** 299  
**D** 100  
**E** 200  
**F** kód obsahuje syntaktickou chybu
- 11** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i <= 3; i++)`  
`y = y + x[i];`  
**A** 6  
**B** 5  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 3  
**E** 9

**12** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A** 101
- B** 103
- C** kód obsahuje syntaktickou chybu
- D** 102

**13** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = &pA + 200;
Bude obsah proměnné A roven:
```

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** 200
- C** 300
- D** kód obsahuje syntaktickou chybu
- E** 100

**14** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A** kód se nepřeloží (obsahuje chybu)
- B** "!"
- C** "Hello World! , Hello World!"
- D** "Hello World!"
- E** "Hello, Hello World!!"
- F** " , Hello World!!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 95           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole o 8 znacích a textový řetězec o délce 8 znaků
- C** Pole o 7 znacích a textový řetězec o délce 8 znaků
- D** Pole o 8 znacích a textový řetězec o délce 7 znaků
- E** Pole o 7 znacích a textový řetězec o délce 7 znaků

**2** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 9
- B** kód se nepřeloží (obsahuje chybu)
- C** 6
- D** 3
- E** 5

**3** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {`  
`}`  
proběhne:

- A** 0x
- B** kód se nepřeloží (obsahuje chybu)
- C** 100x
- D** 99x
- E** 101x

**4** Výsledkem výrazu `(1 || A)`, kde A je proměnná typu integer je:

- A** 0 pokud je A rovno 0
- B** není definováno
- C** 0
- D** nenulová hodnota

**5** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
Bude obsah proměnné A roven:
```

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** kód obsahuje syntaktickou chybu
- C** 200
- D** 100
- E** 300

**6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```

- A** -100
- B** 1
- C** kód se nepřeloží (obsahuje chybu)
- D** 0
- E** -99

**7** Následující zápis reprezentuje:  
`int x[10];`

- A** Pole s 10 prvky s indexy 1..10 typu int
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 10 prvky s indexy 0..9 typu int
- D** Pole s 9 prvky s indexy 0..9 typu int
- E** Pole s 9 prvky s indexy 1..9 typu int

**8** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello World! , Hello World!"
- B** kód se nepřeloží (obsahuje chybu)
- C** "Hello World!"
- D** "!"
- E** "Hello World! , Hello World!!"
- F** "Hello, Hello World!!"

**9** Datový typ char je:

- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B** Typ pro uchovávání velkých celých čísel
- C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D** Typ pro uchovávání malých celých čísel nebo znaků

**10** Cyklus  
`int i = 0;`  
`while (i <= 100) {`  
 `i++;`  
`}`  
proběhne:

- A** kód obsahuje syntaktickou chybu
- B** 99x
- C** 101x
- D** 100x

- 11** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** cyklus nemusí proběhnout ani jednou  
**B** vždy 100x  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu

- 12** Po provedení kódu:  
int A = 100;  
A = --A + 200;  
Bude obsah proměné A roven:  
**A** 299  
**B** 100  
**C** 300  
**D** 200  
**E** 101  
**F** kód obsahuje syntaktickou chybu

- 13** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 1  
**B** 2  
**C** 10  
**D** 6  
**E** 0

- 14** Po provedení kódu:  
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměné A roven:  
**A** 100  
**B** 101  
**C** kód obsahuje syntaktickou chybu  
**D** 98  
**E** 102

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 96           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Datový typ int má vždy velikost:

- A v závislosti na architektuře
- B 8 bajty
- C 2 bajty
- D 4 bajty

**2** Po provedení kódu:

```
int A = 100;  
A = A++ + 200;  
Bude obsah proměnné A roven:
```

- A 301
- B 200
- C kód obsahuje syntaktickou chybu
- D 100
- E 300
- F 101

**3** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}  
bude obsah proměnné A roven:
```

- A 98
- B kód obsahuje syntaktickou chybu
- C 100
- D 97
- E 101

**4** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World! , Hello World!"
- B "Hello World!"
- C " , Hello World!!"
- D "!"
- E "Hello, Hello World!!"
- F kód se nepřeloží (obsahuje chybu)

**5** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A 0
- B kód se nepřeloží (obsahuje chybu)
- C 101
- D 1
- E -1

**6** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A 2
- B -1.8
- C -1
- D 1
- E 0

**7** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 9 prvky s indexy 0..9 typu double
- B Pole s 9 prvky s indexy 1..9 typu double
- C Pole s 10 prvky s indexy 0..9 typu double
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 1..10 typu double

**8** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:

- A 0
- B 10
- C není definováno
- D 1

**9** Cyklus

```
int i = 99;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A dvakrát
- B ani jednou
- C kód obsahuje syntaktickou chybu
- D jednou

- 10** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {`  
`}`  
proběhne:  
**A** 99x  
**B** 0x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 100x  
**E** 101x
- 
- 11** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
`i++;`  
`}`  
proběhne:  
**A** 99x  
**B** 100x  
**C** kód obsahuje syntaktickou chybu  
**D** 101x
- 
- 12** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i < 3; i++)`  
`y = y + x[i];`  
**A** 5  
**B** 9  
**C** 3  
**D** 6  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *pA + 200;`  
Bude obsah proměnné A roven:  
**A** 300  
**B** 100  
**C** kód obsahuje syntaktickou chybu  
**D** 200  
**E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 
- 14** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`  
**A** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 97           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello, Hello World!!"
- B "!"
- C "Hello World! , Hello World!!"
- D kód se nepřeloží (obsahuje chybu)
- E "Hello World!"
- F "Hello World! , Hello World!"

**2** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 7 znaků
- B Pole o 8 znacích a textový řetězec o délce 7 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 7 znacích a textový řetězec o délce 8 znaků
- E Pole o 8 znacích a textový řetězec o délce 8 znaků

**3** Cyklus  

```
do {  
    i++;  
} while (i < 100);
```

  
kde i je proměná typu int proběhne:

- A cyklus nemusí proběhnout ani jednou
- B alespoň jednou
- C vždy 100x
- D kód obsahuje syntaktickou chybu

**4** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A 101
- B -1
- C kód se nepřeloží (obsahuje chybu)
- D 1
- E 0

**5** Datový typ int má vždy velikost:

- A v závislosti na architektuře
- B 8 bajty
- C 2 bajty
- D 4 bajty

**6** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 60
- B 40
- C 30
- D 4
- E 0

**7** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
proběhne:
```

- A 0x
- B 100x
- C kód se nepřeloží (obsahuje chybu)
- D 101x
- E 99x

**8** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

  
Bude obsah proměně A roven:

- A 200
- B 300
- C 100
- D obsah hodnoty A nelze bez spuštění jednoznačně určit
- E kód obsahuje syntaktickou chybu

**9** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 9
- B 5
- C kód se nepřeloží (obsahuje chybu)
- D 3
- E 6

**10** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}
```

  
bude obsah proměně A roven:

- A 102
- B 103
- C kód obsahuje syntaktickou chybu
- D 101

- 11** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
    `i++;`  
`}`  
proběhne:  
**A** 101x  
**B** 99x  
**C** kód obsahuje syntaktickou chybu  
**D** 100x

- 12** Výsledkem výrazu `(1 && 1)` je:  
**A** 10  
**B** není definováno  
**C** 0  
**D** 1

- 13** Po provedení kódu:  
`int A = 100;`  
`A = --A + 200;`  
Bude obsah proměnné A roven:  
**A** 100  
**B** kód obsahuje syntaktickou chybu  
**C** 299  
**D** 300  
**E** 200  
**F** 101

- 14** Následující zápis reprezentuje:  
`int x[10];`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 10 prvky s indexy 0..9 typu int  
**C** Pole s 10 prvky s indexy 1..10 typu int  
**D** Pole s 9 prvky s indexy 1..9 typu int  
**E** Pole s 9 prvky s indexy 0..9 typu int

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 98           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Cyklus  

```
int i = 99;  
do {  
    i++;  
} while (i < 100);
```

  
proběhne:

- A jednou
- B dvakrát
- C kód obsahuje syntaktickou chybu
- D ani jednou

**2** Datový typ float je:

- A Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- B Typ se znaménkem pro uchovávání celých čísel
- C Typ bez znaménka pro uchovávání celých čísel
- D Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**3** Následující zápis reprezentuje:  

```
double x[10];
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole s 10 prvků s indexy 1..10 typu double
- C Pole s 10 prvků s indexy 0..9 typu double
- D Pole s 9 prvků s indexy 1..9 typu double
- E Pole s 9 prvků s indexy 0..9 typu double

**4** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 7 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 7 znacích a textový řetězec o délce 8 znaků
- D Pole o 8 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 7 znaků

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 3
- C 9
- D 6
- E 5

**6** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}
```

  
bude obsah proměně A roven:

- A 98
- B 101
- C 100
- D 97
- E kód obsahuje syntaktickou chybu

**7** Po provedení kódu:  

```
int A = 100;  
A = A++ + 200;
```

  
Bude obsah proměně A roven:

- A 101
- B kód obsahuje syntaktickou chybu
- C 300
- D 100
- E 200
- F 301

**8** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

  
kde i je proměnná typu int proběhne:

- A alespoň jednou
- B vždy 100x
- C kód obsahuje syntaktickou chybu
- D cyklus nemusí proběhnout ani jednou

**9** Výsledkem výrazu (1 && 1) je:

- A 10
- B 1
- C není definováno
- D 0

**10** Hodnota výrazu (3/2)\*(10+10)\*2 je:

- A 40
- B 60
- C 30
- D 4
- E 0

**11** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "Hello World!"
- C "Hello, Hello World!!"
- D "Hello World! , Hello World!"
- E "!"
- F " , Hello World!!"

**12** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A 1
- B 101
- C 0
- D kód se nepřeloží (obsahuje chybu)
- E -1

**13** Tělo cyklu  
for (int i = 0; i < 100; i++) {  
}  
proběhne:

- A 0x
- B 101x
- C 100x
- D 99x
- E kód se nepřeloží (obsahuje chybu)

**14** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = pA + 200;
Bude obsah proměnné A roven:
```

- A obsah hodnoty A nelze bez spuštění jednoznačně určit
- B 100
- C kód obsahuje syntaktickou chybu
- D 200
- E 300

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 99           |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 10 prvky s indexy 0..9 typu double  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 9 prvky s indexy 1..9 typu double  
**E** Pole s 10 prvky s indexy 1..10 typu double
- 2** Cyklus  
`int i = 99;`  
`do {`  
`i++;`  
`} while (i < 100);`  
proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** dvakrát  
**C** jednou  
**D** ani jednou
- 3** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** " , Hello World!!"  
**B** "!"  
**C** "Hello, Hello World!!"  
**D** "Hello World!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "Hello World! , Hello World!"
- 4** Datový typ float je:
- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**B** Typ bez znaménka pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání celých čísel
- 5** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:
- A** 10  
**B** 0  
**C** 1  
**D** není definováno
- 6** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**D** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 7 znaků

- 7** Cyklus  
`while (i < 100) {`  
`i++;`  
`}`  
kde i je proměnná typu int proběhne:
- A** alespoň jednou  
**B** vždy 100x  
**C** kód obsahuje syntaktickou chybu  
**D** cyklus nemusí proběhnout ani jednou
- 8** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *pA + 200;`  
Bude obsah proměnné A roven:
- A** 300  
**B** 100  
**C** kód obsahuje syntaktickou chybu  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 200
- 9** Po provedení kódu:  
`int A = 100;`  
`switch (A) {`  
`case 100: A++;`  
`case 101: A++; break;`  
`case 102: A++;`  
`}`  
bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 102  
**C** 101  
**D** 103
- 10** Tělo cyklu  
`for (int i = 0; i < 100; i++) {`  
`}`  
proběhne:
- A** 99x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 100x  
**D** 101x  
**E** 0x
- 11** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:
- A** 0  
**B** 40  
**C** 60  
**D** 30  
**E** 4

- 12** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 100
  - C** 200
  - D** kód obsahuje syntaktickou chybu
  - E** 300
  - F** 301
- 
- 13** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x += 1; }`
- A** kód se nepřeloží (obsahuje chybu)
  - B** 100
  - C** 99
  - D** 0
  - E** 1
- 
- 14** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };`  
`int y = 0;`  
`for (int i = 1; i <= 3; i++)`  
`y = y + x[i];`
- A** 3
  - B** 5
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 6
  - E** 9

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 100          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** 3  
**B** 9  
**C** 6  
**D** 5  
**E** kód se nepřeloží (obsahuje chybu)

- 2** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** 100  
**B** 101  
**C** 102  
**D** kód obsahuje syntaktickou chybu

- 3** Hodnota výrazu  $(2+3)/5+5$  je:
- A** 0  
**B** 6  
**C** 2  
**D** 10  
**E** 1

- 4** Po provedení kódu:
- ```
int A = 100;
A = A++ + 200;
```
- Bude obsah proměnné A roven:
- A** 100  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** 101  
**E** 301  
**F** 200

- 5** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:
- A** 1  
**B** 0  
**C** 10  
**D** není definováno

- 6** Datový typ int má vždy velikost:
- A** 4 bajty  
**B** v závislosti na architektuře  
**C** 2 bajty  
**D** 8 bajty

- 7** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = &pA + 200;
```
- Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 300  
**C** 200  
**D** 100  
**E** obsah hodnoty A nelze bez spuštění jednoznačně určit

- 8** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** 99x  
**B** 100x  
**C** kód obsahuje syntaktickou chybu  
**D** 101x

- 9** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 10 prvky s indexy 0..9 typu double  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** Pole s 9 prvky s indexy 1..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 1..10 typu double

- 10** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** 1  
**B** -1  
**C** 0  
**D** 101  
**E** kód se nepřeloží (obsahuje chybu)

- 11** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");`
- A** kód se nepřeloží (obsahuje chybu)  
**B** " , Hello World!!"  
**C** "Hello, Hello World!"  
**D** "!"  
**E** "Hello World!"  
**F** "Hello World! , Hello World!"
- 
- 12** Cyklus  
`int i = 99;  
do {  
 i++;  
} while (i < 100);`  
proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** jednou  
**C** dvakrát  
**D** ani jednou
- 
- 13** Tělo cyklu  
`for (int i = 1; i <= 100; i++) {  
}`  
proběhne:
- A** kód se nepřeloží (obsahuje chybu)  
**B** 99x  
**C** 0x  
**D** 100x  
**E** 101x
- 
- 14** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 7 znacích a textový řetězec o délce 8 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 8 znacích a textový řetězec o délce 8 znaků



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		101

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - E** Pole o 6 znacích a textový řetězec o délce 6 znaků

- 2** Následující zápis reprezentuje:  
`int x[10];`
- A** Pole s 10 prvky s indexy 1..10 typu int
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole s 9 prvky s indexy 1..9 typu int
  - D** Pole s 9 prvky s indexy 0..9 typu int
  - E** Pole s 10 prvky s indexy 0..9 typu int

- 3** Cyklus  
`while (i < 100) {  
 i++;  
}`  
kde i je proměná typu int proběhne:
- A** cyklus nemusí proběhnout ani jednou
  - B** kód obsahuje syntaktickou chybu
  - C** alespoň jednou
  - D** vždy 100x

- 4** Po provedení kódu:  
`int A = 100;  
A = ++A + 200;`  
Bude obsah proměné A roven:
- A** 200
  - B** kód obsahuje syntaktickou chybu
  - C** 301
  - D** 100
  - E** 300
  - F** 101

- 5** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměné A roven:
- A** 300
  - B** 100
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 200
  - E** kód obsahuje syntaktickou chybu

- 6** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:
- A** 60
  - B** 0
  - C** 30
  - D** 40
  - E** 4

- 7** Cyklus  
`int i = 100;  
do {  
 i++;  
} while (i < 100);`  
proběhne:
- A** ani jednou
  - B** dvakrát
  - C** jednou
  - D** kód obsahuje syntaktickou chybu

- 8** Tělo cyklu  
`for (int i = 1; i < 100; i++) {  
}`  
proběhne:
- A** 101x
  - B** 0x
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 100x
  - E** 99x

- 9** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");`
- A** "Hello World!"
  - B** "Hello World! , Hello World!!"
  - C** "!"
  - D** "Hello, Hello World!!"
  - E** kód se nepřeloží (obsahuje chybu)
  - F** "Hello World! , Hello World!"

- 10** Datový typ char je:
- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - B** Typ pro uchovávání malých celých čísel nebo znaků
  - C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - D** Typ pro uchovávání velkých celých čísel

- 11** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:
- A** 10
  - B** není definováno
  - C** 1
  - D** 0

**12** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
bude obsah proměnné A roven:
```

- A** 100
- B** 98
- C** kód obsahuje syntaktickou chybu
- D** 101
- E** 97

**13** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++){
    { x += 1; }
```

- A** 100
- B** kód se nepřeloží (obsahuje chybu)
- C** 1
- D** 0
- E** 99

**14** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 5
- B** 3
- C** 9
- D** 6
- E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		102

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

**A** 0  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 1  
**D** 99  
**E** 100

- 2** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:

**A** kód obsahuje syntaktickou chybu  
**B** 102  
**C** 101  
**D** 100

- 3** Datový typ char je:
- A** Typ pro uchovávání velkých celých čísel  
**B** Typ pro uchovávání malých celých čísel nebo znaků  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

- 4** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

**A** 6  
**B** 9  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 3  
**E** 5

- 5** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

**A** 0  
**B** 40  
**C** 30  
**D** 60  
**E** 4

- 6** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

**A** "Hello World! , Hello World!"  
**B** "!"  
**C** " , Hello World!!"  
**D** "Hello World!"  
**E** "Hello, Hello World!!"  
**F** kód se nepřeloží (obsahuje chybu)

- 7** Tělo cyklu
- ```
for (int i = 0; i <= 100; i++) {
}
```
- proběhne:

**A** 99x  
**B** 0x  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 101x  
**E** 100x

- 8** Cyklus
- ```
int i = 100;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** ani jednou  
**B** kód obsahuje syntaktickou chybu  
**C** dvakrát  
**D** jednou

- 9** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```
- A** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**B** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 10** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

**A** 0  
**B** nenulová hodnota  
**C** není definováno  
**D** 0 pokud je A rovno 0

**11** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 1..9 typu int
- B** Pole s 9 prvky s indexy 0..9 typu int
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole s 10 prvky s indexy 0..9 typu int
- E** Pole s 10 prvky s indexy 1..10 typu int

**12** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** obsah hodnoty A nelze bez spuštění jednoznačně určit
- C** 300
- D** 100
- E** kód obsahuje syntaktickou chybu

**13** Cyklus

```
int i = 0;
```

```
while (i < 100) {
```

```
    i++;
```

```
}
```

proběhne:

- A** kód obsahuje syntaktickou chybu
- B** 100x
- C** 99x
- D** 101x

**14** Po provedení kódu:

```
int A = 100;
```

```
A = ++A + 200;
```

Bude obsah proměnné A roven:

- A** 100
- B** kód obsahuje syntaktickou chybu
- C** 300
- D** 101
- E** 200
- F** 301

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		103

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
    }  
proběhne:
```

  
**A** 101x  
**B** 0x  
**C** 99x  
**D** 100x  
**E** kód se nepřeloží (obsahuje chybu)

- 2** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:  
**A** 1  
**B** 0  
**C** není definováno  
**D** 10

- 3** Po provedení kódu:  

```
int A = 100;  
A = A-- + 200;  
Bude obsah proměné A roven:
```

  
**A** kód obsahuje syntaktickou chybu  
**B** 200  
**C** 100  
**D** 299  
**E** 101  
**F** 300

- 4** Cyklus  

```
int i = 99;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

  
**A** dvakrát  
**B** kód obsahuje syntaktickou chybu  
**C** ani jednou  
**D** jednou

- 5** Datový typ float je:  
**A** Typ se znaménkem pro uchovávání celých čísel  
**B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**D** Typ bez znaménka pro uchovávání celých čísel

- 6** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

  
**A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků

- 7** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;  
Bude obsah proměné A roven:
```

  
**A** 100  
**B** kód obsahuje syntaktickou chybu  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** 200  
**E** 300

- 8** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

  
**A** "Hello World!"  
**B** "Hello, Hello World!!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "!"  
**E** "Hello World! , Hello World!!"  
**F** "Hello World! , Hello World!"

- 9** Následující zápis reprezentuje:  

```
int x[10];
```

  
**A** Pole s 9 prvky s indexy 0..9 typu int  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 1..10 typu int  
**D** Pole s 9 prvky s indexy 1..9 typu int  
**E** Pole s 10 prvky s indexy 0..9 typu int

- 10** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 66  
**B** 1  
**C** 0  
**D** 33  
**E** 44

- 11** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}  
proběhne:
```

  
**A** kód obsahuje syntaktickou chybu  
**B** 99x  
**C** 101x  
**D** 100x

- 12** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** 1  
**B** 101  
**C** 0  
**D** kód se nepřeloží (obsahuje chybu)  
**E** -1

- 13** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```
- bude obsah proměnné A roven:
- A** 103  
**B** kód obsahuje syntaktickou chybu  
**C** 101  
**D** 102

- 14** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 5  
**B** 3  
**C** 6  
**D** 9  
**E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 104          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2+3)/5+5$  je:

- A 10
- B 0
- C 6
- D 1
- E 2

**2** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {`  
`}`  
proběhne:

- A 99x
- B 100x
- C 101x
- D 0x
- E kód se nepřeloží (obsahuje chybu)

**3** Datový typ `int` je:

- A Typ bez znaménka pro uchovávání celých čísel
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ se znaménkem pro uchovávání celých čísel

**4** Cyklus  
`while (i < 100) {`  
`i++;`  
`}`  
kde `i` je proměná typu `int` proběhne:

- A kód obsahuje syntaktickou chybu
- B alespoň jednou
- C vždy 100x
- D cyklus nemusí proběhnout ani jednou

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A 9
- B 5
- C kód se nepřeloží (obsahuje chybu)
- D 6
- E 3

**6** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 6 znacích a textový řetězec o délce 5 znaků
- C Pole o 5 znacích a textový řetězec o délce 6 znaků
- D Pole o 5 znacích a textový řetězec o délce 5 znaků
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

**7** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
Bude obsah proměnné A roven:
```

- A 200
- B kód obsahuje syntaktickou chybu
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D 300
- E 100

**8** Po vykonání následujícího kódu bude v `buf` uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A kód se nepřeloží (obsahuje chybu)
- B "!"
- C "Hello World! , Hello World!"
- D "Hello, Hello World!!"
- E "Hello World! , Hello World!!"
- F "Hello World!"

**9** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A není definováno
- B 1
- C 0
- D 10

**10** Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu `int`
- B Pole s 10 prvky s indexy 1..10 typu `int`
- C Pole s 9 prvky s indexy 0..9 typu `int`
- D kód se nepřeloží (obsahuje chybu)
- E Pole s 10 prvky s indexy 0..9 typu `int`

**11** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné `A` roven:

- A 101
- B 100
- C 102
- D kód obsahuje syntaktickou chybu

**12** Cyklus  
`int i = 99;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:

- A** kód obsahuje syntaktickou chybu
- B** dvakrát
- C** jednou
- D** ani jednou

**13** Hodnota proměnné `x` po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** -1
- B** 1
- C** 0
- D** kód se nepřeloží (obsahuje chybu)
- E** 101

**14** Po provedení kódu:

```
int A = 100;
A = --A + 200;
```

Bude obsah proměnné `A` roven:

- A** 299
- B** 300
- C** 101
- D** 200
- E** 100
- F** kód obsahuje syntaktickou chybu



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 105          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 0
- B 44
- C 1
- D 66
- E 33

**2** Po provedení kódu:

```
int A = 100;
```

```
A = --A + 200;
```

Bude obsah proměněné A roven:

- A 300
- B kód obsahuje syntaktickou chybu
- C 101
- D 299
- E 100
- F 200

**3** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 10 prvky s indexy 0..9 typu double
- B Pole s 9 prvky s indexy 0..9 typu double
- C Pole s 10 prvky s indexy 1..10 typu double
- D Pole s 9 prvky s indexy 1..9 typu double
- E kód se nepřeloží (obsahuje chybu)

**4** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A 0
- B není definováno
- C nenulová hodnota
- D 0 pokud je A rovno 0

**5** Datový typ int má vždy velikost:

- A v závislosti na architektuře
- B 8 bajty
- C 2 bajty
- D 4 bajty

**6** Cyklus

```
int i = 100;
```

```
do {
```

```
    i++;
```

```
} while (i < 100);
```

proběhne:

- A dvakrát
- B jednou
- C kód obsahuje syntaktickou chybu
- D ani jednou

**7** Po provedení kódu:

```
int A = 98;
```

```
switch (A) {
```

```
    case 99: A++; break;
```

```
    case 100: A++; break;
```

```
    case 101: A++; break;
```

```
    default: A--;
```

```
}
```

bude obsah proměněné A roven:

- A 97
- B 101
- C 98
- D kód obsahuje syntaktickou chybu
- E 100

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
```

```
for (int i = 0; i < 100; i++);
```

```
{ x += 1; }
```

- A 1
- B 0
- C kód se nepřeloží (obsahuje chybu)
- D 99
- E 100

**9** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {
```

```
}
```

proběhne:

- A 100x
- B 99x
- C kód se nepřeloží (obsahuje chybu)
- D 101x
- E 0x

**10** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 8 znaků
- B Pole o 7 znacích a textový řetězec o délce 8 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 7 znacích a textový řetězec o délce 7 znaků
- E Pole o 8 znacích a textový řetězec o délce 7 znaků

- 11** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");`
- A** "Hello World!"  
**B** "Hello World! , Hello World!"  
**C** "!"  
**D** "Hello World! , Hello World!!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "Hello, Hello World!!"
- 
- 12** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = &pA + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 200  
**D** 300  
**E** 100
- 
- 13** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
 y = y + x[i];`
- A** kód se nepřeloží (obsahuje chybu)  
**B** 3  
**C** 9  
**D** 6  
**E** 5
- 
- 14** Cyklus  
`while (i < 100) {  
 i++;  
}`  
kde i je proměnná typu int proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** alespoň jednou  
**C** vždy 100x  
**D** cyklus nemusí proběhnout ani jednou

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 106          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
}  
proběhne:
```

- A kód se nepřeloží (obsahuje chybu)
- B 101x
- C 100x
- D 99x
- E 0x

**2** Následující zápis reprezentuje:  

```
double x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu double
- B kód se nepřeloží (obsahuje chybu)
- C Pole s 10 prvky s indexy 1..10 typu double
- D Pole s 10 prvky s indexy 0..9 typu double
- E Pole s 9 prvky s indexy 0..9 typu double

**3** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}  
proběhne:
```

- A kód obsahuje syntaktickou chybu
- B 99x
- C 100x
- D 101x

**4** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x -= 1; }
```

- A 0
- B 1
- C -99
- D kód se nepřeloží (obsahuje chybu)
- E -100

**5** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 8 znacích a textový řetězec o délce 7 znaků
- C Pole o 7 znacích a textový řetězec o délce 7 znaků
- D Pole o 7 znacích a textový řetězec o délce 8 znaků
- E Pole o 8 znacích a textový řetězec o délce 8 znaků

**6** Datový typ int má vždy velikost:

- A 4 bajty
- B 8 bajty
- C 2 bajty
- D v závislosti na architektuře

**7** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 33
- B 1
- C 0
- D 66
- E 44

**8** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:

- A 0
- B 1
- C 10
- D není definováno

**9** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;
```

bude obsah proměnné A roven:

- A 97
- B 101
- C 100
- D kód obsahuje syntaktickou chybu
- E 98

**10** Po provedení kódu:

```
int A = 100;  
A = --A + 200;  
Bude obsah proměnné A roven:
```

- A 200
- B kód obsahuje syntaktickou chybu
- C 100
- D 299
- E 101
- F 300

**11** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A 6
- B 5
- C 3
- D kód se nepřeloží (obsahuje chybu)
- E 9

- 12** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** vždy 100x  
**B** cyklus nemusí proběhnout ani jednou  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu
- 
- 13** Po vykonání následujícího kódu bude v buf uložen řetězec:  
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");  
**A** "Hello World!"  
**B** "Hello World! , Hello World!"  
**C** "!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello, Hello World!!"  
**F** " , Hello World!!"
- 
- 14** Po provedení kódu:  
int A = 100;  
int\* pA = &A;  
A = pA + 200;  
Bude obsah proměné A roven:  
**A** 200  
**B** 300  
**C** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**D** kód obsahuje syntaktickou chybu  
**E** 100

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 107          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu int
- B** Pole s 9 prvky s indexy 1..9 typu int
- C** Pole s 9 prvky s indexy 0..9 typu int
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole s 10 prvky s indexy 1..10 typu int

**2** Tělo cyklu

```
for (int i = 1; i < 100; i++) {  
}
```

proběhne:

- A** 101x
- B** 99x
- C** 100x
- D** 0x
- E** kód se nepřeloží (obsahuje chybu)

**3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** 9
- B** 6
- C** 3
- D** kód se nepřeloží (obsahuje chybu)
- E** 5

**4** Datový typ int má vždy velikost:

- A** v závislosti na architektuře
- B** 2 bajty
- C** 4 bajty
- D** 8 bajty

**5** Po provedení kódu:

```
int A = 100;  
A = --A + 200;  
Bude obsah proměnné A roven:
```

- A** 200
- B** 100
- C** 101
- D** 299
- E** kód obsahuje syntaktickou chybu
- F** 300

**6** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** "Hello World! , Hello World!!"
- B** "Hello, Hello World!!"
- C** "Hello World! , Hello World!"
- D** kód se nepřeloží (obsahuje chybu)
- E** "Hello World!"
- F** "!"

**7** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;  
Bude obsah proměnné A roven:
```

- A** 200
- B** 300
- C** 100
- D** kód obsahuje syntaktickou chybu
- E** obsah hodnoty A nelze bez spuštění jednoznačně určit

**8** Po provedení kódu:

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměnné A roven:
```

- A** 101
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 102

**9** Cyklus

```
int i = 100;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A** ani jednou
- B** jednou
- C** kód obsahuje syntaktickou chybu
- D** dvakrát

- 10** Cyklus  

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:  
**A** 101x  
**B** 100x  
**C** kód obsahuje syntaktickou chybu  
**D** 99x
- 
- 11** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

**A** -1  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 0  
**D** 1  
**E** 101
- 
- 12** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

**A** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:  
**A** 1  
**B** 10  
**C** 0  
**D** není definováno
- 
- 14** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:  
**A** 30  
**B** 40  
**C** 0  
**D** 4  
**E** 60
-

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 108          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;
for (int i = 0; i < 100; i++)
{ x += 1; }
```

- A kód se nepřeloží (obsahuje chybu)
- B 100
- C 1
- D 0
- E 99

**2** Následující zápis reprezentuje:  

```
int x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu int
- B Pole s 10 prvky s indexy 0..9 typu int
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 10 prvky s indexy 1..10 typu int
- E Pole s 9 prvky s indexy 0..9 typu int

**3** Po provedení kódu:  

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 102
- C 100
- D 101
- E 98

**4** Datový typ char je:

- A Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- B Typ pro uchovávání velkých celých čísel
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ pro uchovávání malých celých čísel nebo znaků

**5** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 40
- B 30
- C 0
- D 4
- E 60

**6** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

- A kód se nepřeloží (obsahuje chybu)
- B 100x
- C 99x
- D 101x
- E 0x

**7** Cyklus  

```
int i = 100;
do {
    i++;
} while (i < 100);
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B jednou
- C dvakrát
- D ani jednou

**8** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

- A 0
- B 0 pokud je A rovno 0
- C není definováno
- D 1

**9** Po provedení kódu:  

```
int A = 100;
int* pA = &A;
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 200
- C 100
- D obsah hodnoty A nelze bez spuštění jednoznačně určit
- E 300

**10** Po provedení kódu:  

```
int A = 100;
A = ++A + 200;
```

Bude obsah proměnné A roven:

- A 101
- B kód obsahuje syntaktickou chybu
- C 300
- D 200
- E 301
- F 100

**11** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A "Hello, Hello World!!"
- B "Hello World! , Hello World!"
- C "Hello World! , Hello World!!"
- D "Hello World!"
- E kód se nepřeloží (obsahuje chybu)
- F "!"

**12** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 6 znacích a textový řetězec o délce 6 znaků
- C Pole o 5 znacích a textový řetězec o délce 6 znaků
- D Pole o 5 znacích a textový řetězec o délce 5 znaků
- E Pole o 6 znacích a textový řetězec o délce 5 znaků

**13** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A 99x
- B kód obsahuje syntaktickou chybu
- C 100x
- D 101x

**14** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 9
- B 6
- C kód se nepřeloží (obsahuje chybu)
- D 5
- E 3



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 109          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměné A roven:

- A 101
- B kód obsahuje syntaktickou chybu
- C 102
- D 103

**2** Po provedení kódu:

```
int A = 100;
A = A++ + 200;
Bude obsah proměné A roven:
```

- A 300
- B 101
- C 100
- D kód obsahuje syntaktickou chybu
- E 200
- F 301

**3** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
Bude obsah proměné A roven:
```

- A 100
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C kód obsahuje syntaktickou chybu
- D 300
- E 200

**4** Cyklus

```
int i = 100;
do {
    i++;
} while (i < 100);
proběhne:
```

- A ani jednou
- B dvakrát
- C jednou
- D kód obsahuje syntaktickou chybu

**5** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {
}
```

proběhne:

- A kód se nepřeloží (obsahuje chybu)
- B 99x
- C 101x
- D 100x
- E 0x

**6** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A 100x
- B 101x
- C kód obsahuje syntaktickou chybu
- D 99x

**7** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

- A 1
- B kód se nepřeloží (obsahuje chybu)
- C 100
- D 99
- E 0

**8** Datový typ float je:

- A Typ se znaménkem pro uchovávání celých čísel
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D Typ bez znaménka pro uchovávání celých čísel

**9** Následující zápis reprezentuje:

```
double x[10];
```

- A Pole s 10 prvky s indexy 1..10 typu double
- B Pole s 10 prvky s indexy 0..9 typu double
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 9 prvky s indexy 1..9 typu double
- E Pole s 9 prvky s indexy 0..9 typu double

**10** Výsledkem výrazu (1 && 1) je:

- A 10
- B 1
- C není definováno
- D 0

- 11** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** 9  
**C** 6  
**D** 3  
**E** 5
- 
- 12** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků
- 
- 13** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 66  
**B** 1  
**C** 33  
**D** 44  
**E** 0
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "!"  
**B** "Hello, Hello World!!"  
**C** "Hello World!"  
**D** "Hello World! , Hello World!!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "Hello World! , Hello World!"

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		110

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Tělo cyklu  

```
for (int i = 1; i < 100; i++) {  
}  
proběhne:
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** 0x  
**C** 101x  
**D** 99x  
**E** 100x
- 2** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
    default: A--;  
}  
bude obsah proměné A roven:
```

**A** 100  
**B** 97  
**C** 98  
**D** kód obsahuje syntaktickou chybu  
**E** 101
- 3** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 44  
**B** 0  
**C** 66  
**D** 1  
**E** 33
- 4** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:  
**A** 0  
**B** není definováno  
**C** 0 pokud je A rovno 0  
**D** 1
- 5** Následující zápis reprezentuje:  

```
double x[10];
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** Pole s 10 prvky s indexy 1..10 typu double  
**D** Pole s 9 prvky s indexy 1..9 typu double  
**E** Pole s 10 prvky s indexy 0..9 typu double
- 6** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

**A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**E** kód se nepřeloží (obsahuje chybu)
- 7** Datový typ int má vždy velikost:  
**A** v závislosti na architektuře  
**B** 4 bajty  
**C** 8 bajty  
**D** 2 bajty
- 8** Po provedení kódu:  

```
int A = 100;  
A = ++A + 200;
```

Bude obsah proměné A roven:  
**A** 101  
**B** 200  
**C** 300  
**D** 301  
**E** 100  
**F** kód obsahuje syntaktickou chybu
- 9** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}  
proběhne:
```

**A** 100x  
**B** 101x  
**C** 99x  
**D** kód obsahuje syntaktickou chybu
- 10** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

**A** 6  
**B** 3  
**C** 9  
**D** 5  
**E** kód se nepřeloží (obsahuje chybu)
- 11** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

**A** kód se nepřeloží (obsahuje chybu)  
**B** -1  
**C** 101  
**D** 0  
**E** 1

- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = pA + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** 200
  - C** kód obsahuje syntaktickou chybu
  - D** 100
  - E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 
- 13** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcpy(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** kód se nepřeloží (obsahuje chybu)
  - B** "!"
  - C** " , Hello World!!"
  - D** "Hello World! , Hello World!"
  - E** "Hello World!"
  - F** "Hello, Hello World!!"
- 
- 14** Cyklus  
`do {`  
    `i++;`  
`} while (i < 100);`  
kde i je proměnná typu int proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** alespoň jednou
  - C** vždy 100x
  - D** cyklus nemusí proběhnout ani jednou

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		<b>111</b>

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 5 znacích a textový řetězec o délce 6 znaků
- B** Pole o 6 znacích a textový řetězec o délce 6 znaků
- C** Pole o 5 znacích a textový řetězec o délce 5 znaků
- D** Pole o 6 znacích a textový řetězec o délce 5 znaků
- E** kód se nepřeloží (obsahuje chybu)

**2** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 5
- C** 6
- D** 9
- E** 3

**3** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A** 102
- B** 103
- C** 101
- D** kód obsahuje syntaktickou chybu

**4** Po provedení kódu:

```
int A = 100;
A = A++ + 200;
Bude obsah proměnné A roven:
```

- A** 301
- B** 101
- C** 300
- D** 100
- E** kód obsahuje syntaktickou chybu
- F** 200

**5** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

- A** 99x
- B** kód se nepřeloží (obsahuje chybu)
- C** 101x
- D** 100x
- E** 0x

**6** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A** 2
- B** -1
- C** 1
- D** -1.8
- E** 0

**7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello, Hello World!!"
- B** "Hello World! , Hello World!!"
- C** "!"
- D** "Hello World! , Hello World!"
- E** kód se nepřeloží (obsahuje chybu)
- F** "Hello World!"

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** 1
- B** -1
- C** 0
- D** 101
- E** kód se nepřeloží (obsahuje chybu)

**9** Následující zápis reprezentuje:

```
double x[10];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** Pole s 9 prvky s indexy 1..9 typu double
- C** Pole s 10 prvky s indexy 1..10 typu double
- D** Pole s 9 prvky s indexy 0..9 typu double
- E** Pole s 10 prvky s indexy 0..9 typu double

**10** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A** 99x
- B** 100x
- C** 101x
- D** kód obsahuje syntaktickou chybu

- 
- 11** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** nenulová hodnota
  - B** 0 pokud je A rovno 0
  - C** není definováno
  - D** 0
- 
- 12** Po provedení kódu:
- ```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```
- Bude obsah proměnné A roven:
- A** 300
  - B** 100
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** kód obsahuje syntaktickou chybu
  - E** 200
- 
- 13** Datový typ int je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ se znaménkem pro uchovávání celých čísel
  - C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- 
- 14** Cyklus
- ```
int i = 0;  
do {  
    i++;  
} while (i < 100);
```
- proběhne:
- A** 99x
  - B** kód obsahuje syntaktickou chybu
  - C** 100x
  - D** 101x
-

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		112

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** 1  
**B** 101  
**C** 0  
**D** -1  
**E** kód se nepřeloží (obsahuje chybu)

- 2** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** Pole s 10 prvky s indexy 1..10 typu double  
**B** Pole s 9 prvky s indexy 1..9 typu double  
**C** Pole s 10 prvky s indexy 0..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 9 prvky s indexy 0..9 typu double

- 3** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World!"  
**B** "!"  
**C** "Hello World! , Hello World!!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "Hello, Hello World!!"  
**F** "Hello World! , Hello World!"

- 4** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A** 98  
**B** 101  
**C** 102  
**D** kód obsahuje syntaktickou chybu  
**E** 100

- 5** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A** 3  
**B** 5  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 6  
**E** 9

- 6** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**E** kód se nepřeloží (obsahuje chybu)

- 7** Tělo cyklu
- ```
for (int i = 1; i < 100; i++) {
}
```
- proběhne:
- A** 0x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 101x  
**D** 99x  
**E** 100x

- 8** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** 99x  
**C** 101x  
**D** 100x

- 9** Cyklus
- ```
int i = 0;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** 99x  
**C** 101x  
**D** 100x

**10** Datový typ `int` má vždy velikost:

- A** 8 bajty
- B** 4 bajty
- C** v závislosti na architektuře
- D** 2 bajty

**11** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = **pA + 200;
```

Bude obsah proměnné `A` roven:

- A** 200
- B** obsah hodnoty `A` nelze bez spuštění jednoznačně určit
- C** kód obsahuje syntaktickou chybu
- D** 300
- E** 100

**12** Po provedení kódu:

```
int A = 100;
```

```
A = ++A + 200;
```

Bude obsah proměnné `A` roven:

- A** 300
- B** 301
- C** 101
- D** 200
- E** 100
- F** kód obsahuje syntaktickou chybu

**13** Výsledkem výrazu `(1 && 1)` je:

- A** 10
- B** 1
- C** 0
- D** není definováno

**14** Hodnota výrazu `(2+3)/5+5` je:

- A** 10
- B** 2
- C** 1
- D** 6
- E** 0



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 113          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující zápis reprezentuje:  
`double x[10];`
- A** Pole s 9 prvky s indexy 1..9 typu double  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 9 prvky s indexy 0..9 typu double  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** Pole s 10 prvky s indexy 1..10 typu double

- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!!");`
- A** "Hello, Hello World!!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** "Hello World! , Hello World!"  
**D** "!"  
**E** "Hello World! , Hello World!!"  
**F** "Hello World!"

- 3** Cyklus  
`int i = 100;  
do {  
 i++;  
} while (i < 100);`  
proběhne:
- A** ani jednou  
**B** kód obsahuje syntaktickou chybu  
**C** jednou  
**D** dvakrát

- 4** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
if (1)  
 x = x + 1;  
else  
 x = x - 1;  
for (int i = 0; i < 100; i++) {x++};`
- A** -1  
**B** 1  
**C** 0  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 101

- 5** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = *A + 200;`  
Bude obsah proměnné A roven:
- A** 100  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 200

- 6** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků

- 7** Po provedení kódu:  
`int A = 100;  
switch (A) {  
 case 100: A++;  
 case 101: A++; break;  
 case 102: A++;  
}`  
bude obsah proměnné A roven:
- A** 103  
**B** kód obsahuje syntaktickou chybu  
**C** 102  
**D** 101

- 8** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
 y = y + x[i];`
- A** 6  
**B** 3  
**C** 5  
**D** 9  
**E** kód se nepřeloží (obsahuje chybu)

- 9** Cyklus  

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:  
**A** 99x  
**B** 100x  
**C** kód obsahuje syntaktickou chybu  
**D** 101x

- 10** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {
}
```

proběhne:  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 100x  
**C** 101x  
**D** 0x  
**E** 99x

- 11** Hodnota výrazu  $(2+3)/5+5$  je:  
**A** 0  
**B** 2  
**C** 6  
**D** 10  
**E** 1

- 12** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:  
**A** 0 pokud je A rovno 0  
**B** není definováno  
**C** 1  
**D** 0

- 13** Datový typ int má vždy velikost:  
**A** 4 bajty  
**B** 2 bajty  
**C** v závislosti na architektuře  
**D** 8 bajty

- 14** Po provedení kódu:  

```
int A = 100;
A = A++ + 200;
```

Bude obsah proměnné A roven:  
**A** 300  
**B** 301  
**C** kód obsahuje syntaktickou chybu  
**D** 200  
**E** 100  
**F** 101

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 114          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```
- A** "!"  
**B** "Hello, Hello World!!"  
**C** "Hello World! , Hello World!!"  
**D** "Hello World! , Hello World!"  
**E** kód se nepřeloží (obsahuje chybu)  
**F** "Hello World!"

- 2** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```
- A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 6 znaků

- 3** Cyklus  

```
int i = 0;  
while (i < 100) {  
    i++;  
}
```

  
proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 99x  
**D** 101x

- 4** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```
- A** 3  
**B** 5  
**C** 6  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 9

- 5** Datový typ char je:
- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou  
**B** Typ pro uchovávání velkých celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ pro uchovávání malých celých čísel nebo znaků

- 6** Cyklus  

```
do {  
    i++;  
} while (i < 100);
```

  
kde i je proměnná typu int proběhne:
- A** vždy 100x  
**B** alespoň jednou  
**C** cyklus nemusí proběhnout ani jednou  
**D** kód obsahuje syntaktickou chybu

- 7** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 33  
**B** 44  
**C** 66  
**D** 0  
**E** 1

- 8** Následující zápis reprezentuje:  

```
double x[10];
```
- A** Pole s 9 prvky s indexy 1..9 typu double  
**B** Pole s 9 prvky s indexy 0..9 typu double  
**C** Pole s 10 prvky s indexy 0..9 typu double  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 10 prvky s indexy 1..10 typu double

- 9** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;
```

  
Bude obsah proměnné A roven:
- A** 300  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 100  
**D** 200  
**E** kód obsahuje syntaktickou chybu

- 10** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

  
bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 101  
**C** 100  
**D** 98  
**E** 102

**11** Tělo cyklu  
`for (int i = 1; i < 100; i++) {`  
`}`  
proběhne:

- A** kód se nepřeloží (obsahuje chybu)
- B** 101x
- C** 0x
- D** 99x
- E** 100x

**12** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:

- A** 200
- B** 301
- C** 300
- D** 101
- E** kód obsahuje syntaktickou chybu
- F** 100

**13** Výsledkem výrazu `(1 && 1)` je:

- A** 1
- B** není definováno
- C** 10
- D** 0

**14** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

- A** 1
- B** kód se nepřeloží (obsahuje chybu)
- C** 100
- D** 99
- E** 0

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 115          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World!"
- B " , Hello World!!"
- C "!"
- D "Hello, Hello World!!"
- E kód se nepřeloží (obsahuje chybu)
- F "Hello World! , Hello World!"

**2** Po provedení kódu:  

```
int A = 100;  
A = A++ + 200;
```

Bude obsah proměnné A roven:

- A 301
- B 100
- C kód obsahuje syntaktickou chybu
- D 200
- E 101
- F 300

**3** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A 102
- B 100
- C 101
- D kód obsahuje syntaktickou chybu

**4** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A 6
- B kód se nepřeloží (obsahuje chybu)
- C 9
- D 3
- E 5

**5** Následující kód reprezentuje:  

```
char text[] = "Ahoj!";
```

- A kód se nepřeloží (obsahuje chybu)
- B Pole o 5 znacích a textový řetězec o délce 5 znaků
- C Pole o 5 znacích a textový řetězec o délce 6 znaků
- D Pole o 6 znacích a textový řetězec o délce 6 znaků
- E Pole o 6 znacích a textový řetězec o délce 5 znaků

**6** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A 0 pokud je A rovno 0
- B nenulová hodnota
- C není definováno
- D 0

**7** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
}
```

proběhne:

- A 0x
- B 100x
- C kód se nepřeloží (obsahuje chybu)
- D 101x
- E 99x

**8** Následující zápis reprezentuje:  

```
double x[10];
```

- A Pole s 10 prvky s indexy 1..10 typu double
- B Pole s 10 prvky s indexy 0..9 typu double
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 9 prvky s indexy 0..9 typu double
- E Pole s 9 prvky s indexy 1..9 typu double

**9** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A kód obsahuje syntaktickou chybu
- B 100
- C 300
- D obsah hodnoty A nelze bez spuštění jednoznačně určit
- E 200

**10** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 40
- B 4
- C 30
- D 60
- E 0

- 11** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:
- A** alespoň jednou
  - B** kód obsahuje syntaktickou chybu
  - C** vždy 100x
  - D** cyklus nemusí proběhnout ani jednou

- 12** Datový typ char je:
- A** Typ pro uchovávání malých celých čísel nebo znaků
  - B** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - C** Typ pro uchovávání velkých celých čísel
  - D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou

- 13** Cyklus  
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:
- A** kód obsahuje syntaktickou chybu
  - B** 100x
  - C** 99x
  - D** 101x

- 14** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x += 1; }
```
- A** 100
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 99
  - D** 1
  - E** 0

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		116

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 7 znacích a textový řetězec o délce 8 znaků
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole o 8 znacích a textový řetězec o délce 8 znaků
- D** Pole o 7 znacích a textový řetězec o délce 7 znaků
- E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**2** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
{ x -= 1; }
```

- A** -100
- B** kód se nepřeloží (obsahuje chybu)
- C** 1
- D** -99
- E** 0

**3** Datový typ int má vždy velikost:

- A** 8 bajty
- B** 4 bajty
- C** v závislosti na architektuře
- D** 2 bajty

**4** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 9
- B** 3
- C** 5
- D** kód se nepřeloží (obsahuje chybu)
- E** 6

**5** Po provedení kódu:

```
int A = 100;
A = --A + 200;
Bude obsah proměnné A roven:
```

- A** kód obsahuje syntaktickou chybu
- B** 200
- C** 100
- D** 101
- E** 299
- F** 300

**6** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
proběhne:
```

- A** kód obsahuje syntaktickou chybu
- B** 100x
- C** 99x
- D** 101x

**7** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměnné A roven:

- A** 102
- B** 101
- C** kód obsahuje syntaktickou chybu
- D** 103

**8** Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
proběhne:
```

- A** kód obsahuje syntaktickou chybu
- B** 101x
- C** 100x
- D** 99x

**9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
Bude obsah proměnné A roven:
```

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 200
- E** 300

**10** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 10 prvky s indexy 1..10 typu int
- B** kód se nepřeloží (obsahuje chybu)
- C** Pole s 10 prvky s indexy 0..9 typu int
- D** Pole s 9 prvky s indexy 1..9 typu int
- E** Pole s 9 prvky s indexy 0..9 typu int

**11** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** kód se nepřeloží (obsahuje chybu)
- B** "!"
- C** "Hello World!"
- D** "Hello, Hello World!!"
- E** " , Hello World!!"
- F** "Hello World! , Hello World!"

**12** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {  
    }
```

proběhne:

- A** 0x
- B** 101x
- C** 100x
- D** 99x
- E** kód se nepřeloží (obsahuje chybu)

**13** Výsledkem výrazu (1 && 0) je:

- A** 0
- B** není definováno
- C** 10
- D** 1

**14** Hodnota výrazu (2+3)/5+5 je:

- A** 1
- B** 0
- C** 10
- D** 2
- E** 6



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		117

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

1

Cyklus

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:

- A 100x
- B kód obsahuje syntaktickou chybu
- C 101x
- D 99x

2

Datový typ int je:

- A Typ bez znaménka pro uchovávání celých čísel
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ se znaménkem pro uchovávání celých čísel
- D Typ bez znaménka pro uchovávání čísel s desetinou čárkou

3

Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 10 prvky s indexy 1..10 typu int
- B Pole s 10 prvky s indexy 0..9 typu int
- C Pole s 9 prvky s indexy 1..9 typu int
- D Pole s 9 prvky s indexy 0..9 typu int
- E kód se nepřeloží (obsahuje chybu)

4

Výsledkem výrazu (1 && 1) je:

- A 1
- B není definováno
- C 0
- D 10

5

Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

- A ani jednou
- B jednou
- C kód obsahuje syntaktickou chybu
- D dvakrát

6

Po provedení kódu:

```
int A = 100;
switch (A) {
    case 100: A++;
    case 101: A++; break;
    case 102: A++;
}
```

bude obsah proměné A roven:

- A 103
- B 102
- C kód obsahuje syntaktickou chybu
- D 101

7

Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A "Hello, Hello World!!"
- B "Hello World!"
- C " , Hello World!!"
- D "Hello World! , Hello World!"
- E "!"
- F kód se nepřeloží (obsahuje chybu)

8

Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

- A 100x
- B 99x
- C 0x
- D 101x
- E kód se nepřeloží (obsahuje chybu)

9

Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x -= 1; }
```

- A -100
- B 1
- C -99
- D kód se nepřeloží (obsahuje chybu)
- E 0

10

Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A Pole o 8 znacích a textový řetězec o délce 7 znaků
- B Pole o 8 znacích a textový řetězec o délce 8 znaků
- C kód se nepřeloží (obsahuje chybu)
- D Pole o 7 znacích a textový řetězec o délce 8 znaků
- E Pole o 7 znacích a textový řetězec o délce 7 znaků

- 11** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = pA + 200;`  
Bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu
  - B** 300
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 100
  - E** 200

- 12** Po provedení kódu:  
`int A = 100;  
A = --A + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 300
  - C** 200
  - D** 299
  - E** 100
  - F** kód obsahuje syntaktickou chybu

- 13** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 44
  - B** 0
  - C** 33
  - D** 1
  - E** 66

- 14** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```
- A** 6
  - B** 5
  - C** 3
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 9

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 118          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
- B** Pole o 7 znacích a textový řetězec o délce 7 znaků
- C** Pole o 7 znacích a textový řetězec o délce 8 znaků
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**2** Výsledkem výrazu (1 && 0) je:

- A** 1
- B** 0
- C** není definováno
- D** 10

**3** Cyklus

```
int i = 0;
do {
    i++;
} while (i < 100);
proběhne:
```

- A** 99x
- B** kód obsahuje syntaktickou chybu
- C** 101x
- D** 100x

**4** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** -1
- B** 0
- C** kód se nepřeloží (obsahuje chybu)
- D** 101
- E** 1

**5** Po provedení kódu:

```
int A = 100;
A = A++ + 200;
Bude obsah proměnné A roven:
```

- A** 100
- B** 101
- C** 300
- D** 301
- E** kód obsahuje syntaktickou chybu
- F** 200

**6** Tělo cyklu

```
for (int i = 0; i <= 100; i++) {
}
proběhne:
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 0x
- C** 101x
- D** 100x
- E** 99x

**7** Datový typ char je:

- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B** Typ pro uchovávání velkých celých čísel
- C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D** Typ pro uchovávání malých celých čísel nebo znaků

**8** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu int
- B** Pole s 9 prvky s indexy 1..9 typu int
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole s 10 prvky s indexy 1..10 typu int
- E** Pole s 9 prvky s indexy 0..9 typu int

**9** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 102
- C** 101
- D** 100

**10** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello World! , Hello World!"
- B** "Hello, Hello World!!"
- C** "Hello World! , Hello World!!"
- D** "!"
- E** "Hello World!"
- F** kód se nepřeloží (obsahuje chybu)

- 11** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = pA + 200;`  
Bude obsah proměnné A roven:
- A** 200
  - B** kód obsahuje syntaktickou chybu
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 100
  - E** 300

- 12** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
 y = y + x[i];`
- A** 5
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 9
  - D** 3
  - E** 6

- 13** Cyklus  
`int i = 0;  
while (i < 100) {  
 i++;  
}`  
proběhne:
- A** 99x
  - B** 101x
  - C** kód obsahuje syntaktickou chybu
  - D** 100x

- 14** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** -1
  - B** 2
  - C** 1
  - D** -1.8
  - E** 0

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 119          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:
- ```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměné A roven:
- A** 98
  - B** 102
  - C** 101
  - D** kód obsahuje syntaktickou chybu
  - E** 100

- 2** Cyklus
- ```
while (i < 100) {
    i++;
}
```
- kde i je proměná typu int proběhne:
- A** vždy 100x
  - B** alespoň jednou
  - C** cyklus nemusí proběhnout ani jednou
  - D** kód obsahuje syntaktickou chybu

- 3** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměná typu integer je:
- A** 0 pokud je A rovno 0
  - B** nenulová hodnota
  - C** není definováno
  - D** 0

- 4** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```
- A** Pole o 8 znacích a textový řetězec o délce 7 znaků
  - B** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - E** Pole o 7 znacích a textový řetězec o délce 8 znaků

- 5** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** -1.8
  - B** 1
  - C** 2
  - D** -1
  - E** 0

- 6** Po provedení kódu:
- ```
int A = 100;
A = --A + 200;
```
- Bude obsah proměné A roven:
- A** 299
  - B** 101
  - C** 300
  - D** 200
  - E** 100
  - F** kód obsahuje syntaktickou chybu

- 7** Následující zápis reprezentuje:
- ```
double x[10];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole s 10 prvky s indexy 0..9 typu double
  - C** Pole s 9 prvky s indexy 0..9 typu double
  - D** Pole s 10 prvky s indexy 1..10 typu double
  - E** Pole s 9 prvky s indexy 1..9 typu double

- 8** Datový typ int má vždy velikost:
- A** v závislosti na architektuře
  - B** 8 bajty
  - C** 2 bajty
  - D** 4 bajty

- 9** Tělo cyklu
- ```
for (int i = 1; i < 100; i++) {
}
```
- proběhne:
- A** 100x
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 99x
  - D** 101x
  - E** 0x

- 10** Cyklus
- ```
int i = 99;
do {
    i++;
} while (i < 100);
```
- proběhne:
- A** dvakrát
  - B** ani jednou
  - C** jednou
  - D** kód obsahuje syntaktickou chybu

- 11** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A** "Hello World! , Hello World!"  
**B** "Hello, Hello World!!"  
**C** "Hello World! , Hello World!!"  
**D** kód se nepřeloží (obsahuje chybu)  
**E** "!"  
**F** "Hello World!"
- 
- 12** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```
- A** 5  
**B** 6  
**C** 3  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 9
- 
- 13** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = *pA + 200;
```
- Bude obsah proměnné A roven:
- A** 200  
**B** 300  
**C** 100  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** kód obsahuje syntaktickou chybu
- 
- 14** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
for (int i = 0; i < 100; i++)
    { x -= 1; }
```
- A** -99  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 1  
**D** -100  
**E** 0

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		120

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Datový typ `int` je:

  - A** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
  - B** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
  - C** Typ se znaménkem pro uchovávání celých čísel
  - D** Typ bez znaménka pro uchovávání celých čísel
- 2** Hodnota proměnné `x` po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
{ x += 1; }
```

  - A** 99
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 0
  - D** 100
  - E** 1
- 3** Tělo cyklu

```
for (int i = 1; i < 100; i++) {
}
```

proběhne:

  - A** 100x
  - B** 101x
  - C** kód se nepřeloží (obsahuje chybu)
  - D** 0x
  - E** 99x
- 4** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

  - A** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole o 6 znacích a textový řetězec o délce 6 znaků
- 5** Hodnota výrazu  $(2+3)/5+5$  je:

  - A** 2
  - B** 10
  - C** 0
  - D** 1
  - E** 6
- 6** Výsledkem výrazu  $(1 \parallel A)$ , kde `A` je proměnná typu `integer` je:

  - A** 0
  - B** není definováno
  - C** 0 pokud je `A` rovno 0
  - D** nenulová hodnota
- 7** Po provedení kódu:

```
int A = 100;
A = A++ + 200;
```

Bude obsah proměnné `A` roven:

  - A** 300
  - B** 100
  - C** kód obsahuje syntaktickou chybu
  - D** 301
  - E** 101
  - F** 200
- 8** Cyklus

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

  - A** 99x
  - B** 101x
  - C** 100x
  - D** kód obsahuje syntaktickou chybu
- 9** Následující zápis reprezentuje:

```
double x[10];
```

  - A** Pole s 10 prvky s indexy 1..10 typu `double`
  - B** Pole s 10 prvky s indexy 0..9 typu `double`
  - C** kód se nepřeloží (obsahuje chybu)
  - D** Pole s 9 prvky s indexy 0..9 typu `double`
  - E** Pole s 9 prvky s indexy 1..9 typu `double`
- 10** Cyklus

```
int i = 100;
do {
    i++;
} while (i < 100);
```

proběhne:

  - A** ani jednou
  - B** jednou
  - C** kód obsahuje syntaktickou chybu
  - D** dvakrát
- 11** Po vykonání následujícího kódu bude v `buf` uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

  - A** kód se nepřeloží (obsahuje chybu)
  - B** " , Hello World!!"
  - C** "Hello World! , Hello World!"
  - D** "!"
  - E** "Hello, Hello World!!"
  - F** "Hello World!"

**12** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** 9
- B** kód se nepřeloží (obsahuje chybu)
- C** 5
- D** 3
- E** 6

**13** Po provedení kódu:

```
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 100
- C** 102
- D** 101

**14** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = *A + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** obsah hodnoty A nelze bez spuštění jednoznačně určit
- C** kód obsahuje syntaktickou chybu
- D** 300
- E** 100



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		121

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)  
**B** 5  
**C** 9  
**D** 3  
**E** 6

- 2** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků

- 3** Hodnota výrazu  $(2+3)/5+5$  je:

- A** 0  
**B** 2  
**C** 6  
**D** 1  
**E** 10

- 4** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:

- A** 0 pokud je A rovno 0  
**B** 1  
**C** 0  
**D** není definováno

- 5** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = pA + 200;
Bude obsah proměnné A roven:
```

- A** 100  
**B** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**C** 200  
**D** 300  
**E** kód obsahuje syntaktickou chybu

- 6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x += 1; }
```

- A** 99  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 0  
**D** 100  
**E** 1

- 7** Následující zápis reprezentuje:

```
double x[10];
```

- A** Pole s 10 prvky s indexy 0..9 typu double  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole s 10 prvky s indexy 1..10 typu double  
**D** Pole s 9 prvky s indexy 1..9 typu double  
**E** Pole s 9 prvky s indexy 0..9 typu double

- 8** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
```

bude obsah proměnné A roven:

- A** 100  
**B** kód obsahuje syntaktickou chybu  
**C** 98  
**D** 97  
**E** 101

- 9** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello, Hello World!!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** "!"  
**D** " , Hello World!!"  
**E** "Hello World!"  
**F** "Hello World! , Hello World!"

- 10** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** 301
  - C** kód obsahuje syntaktickou chybu
  - D** 100
  - E** 200
  - F** 101

- 11** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
    *i++;*  
`}`  
proběhne:
- A** 99x
  - B** 101x
  - C** 100x
  - D** kód obsahuje syntaktickou chybu

- 12** Datový typ char je:
- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - C** Typ pro uchovávání velkých celých čísel
  - D** Typ pro uchovávání malých celých čísel nebo znaků

- 13** Tělo cyklu  
`for (int i = 0; i < 100; i++) {`  
      
`}`  
proběhne:
- A** kód se nepřeloží (obsahuje chybu)
  - B** 99x
  - C** 101x
  - D** 100x
  - E** 0x

- 14** Cyklus  
`do {`  
    *i++;*  
`} while (i < 100);`  
kde *i* je proměnná typu `int` proběhne:
- A** vždy 100x
  - B** alespoň jednou
  - C** kód obsahuje syntaktickou chybu
  - D** cyklus nemusí proběhnout ani jednou

# PB071: Prubezny test pb071 13:00

Jméno a příjmení - pište do okénka	UČO	Číslo zadání
		122

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 40
- B 4
- C 30
- D 60
- E 0

**2** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:

- A nenulová hodnota
- B není definováno
- C 0
- D 0 pokud je A rovno 0

**3** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 5
- C 3
- D 6
- E 9

**4** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!!");
```

- A "Hello World!"
- B "Hello World! , Hello World!"
- C "!"
- D kód se nepřeloží (obsahuje chybu)
- E "Hello, Hello World!!"
- F "Hello World! , Hello World!!"

**5** Datový typ int má vždy velikost:

- A 4 bajty
- B 8 bajty
- C 2 bajty
- D v závislosti na architektuře

**6** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A 101
- B kód obsahuje syntaktickou chybu
- C 100
- D 102

**7** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
```

Bude obsah proměnné A roven:

- A 299
- B 100
- C kód obsahuje syntaktickou chybu
- D 101
- E 300
- F 200

**8** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++)
    { x += 1; }
```

- A 0
- B 100
- C 99
- D kód se nepřeloží (obsahuje chybu)
- E 1

**9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměnné A roven:

- A obsah hodnoty A nelze bez spuštění jednoznačně určit
- B 200
- C 300
- D kód obsahuje syntaktickou chybu
- E 100

**10** Cyklus  

```
int i = 0;
while (i <= 100) {
    i++;
}
```

proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** 100x  
**C** 101x  
**D** 99x

**11** Následující zápis reprezentuje:  

```
int x[10];
```

  
**A** Pole s 10 prvky s indexy 1..10 typu int  
**B** Pole s 9 prvky s indexy 1..9 typu int  
**C** Pole s 10 prvky s indexy 0..9 typu int  
**D** kód se nepřeloží (obsahuje chybu)  
**E** Pole s 9 prvky s indexy 0..9 typu int

**12** Cyklus  

```
do {
    i++;
} while (i < 100);
```

kde i je proměná typu int proběhne:  
**A** kód obsahuje syntaktickou chybu  
**B** alespoň jednou  
**C** cyklus nemusí proběhnout ani jednou  
**D** vždy 100x

**13** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

  
**A** kód se nepřeloží (obsahuje chybu)  
**B** Pole o 7 znacích a textový řetězec o délce 7 znaků  
**C** Pole o 8 znacích a textový řetězec o délce 7 znaků  
**D** Pole o 8 znacích a textový řetězec o délce 8 znaků  
**E** Pole o 7 znacích a textový řetězec o délce 8 znaků

**14** Tělo cyklu  

```
for (int i = 0; i <= 100; i++) {
}
```

proběhne:  
**A** 99x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 101x  
**D** 0x  
**E** 100x

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		123

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Datový typ `int` má vždy velikost:

- A** 2 bajty
- B** 4 bajty
- C** v závislosti na architektuře
- D** 8 bajty

**2** Cyklus

```
int i = 100;
do {
    i++;
} while (i < 100);
proběhne:
```

- A** ani jednou
- B** kód obsahuje syntaktickou chybu
- C** jednou
- D** dvakrát

**3** Následující zápis reprezentuje:

```
int x[10];
```

- A** Pole s 9 prvky s indexy 0..9 typu `int`
- B** Pole s 9 prvky s indexy 1..9 typu `int`
- C** Pole s 10 prvky s indexy 0..9 typu `int`
- D** Pole s 10 prvky s indexy 1..10 typu `int`
- E** kód se nepřeloží (obsahuje chybu)

**4** Cyklus

```
while (i < 100) {
    i++;
}
kde i je proměná typu int proběhne:
```

- A** vždy 100x
- B** alespoň jednou
- C** kód obsahuje syntaktickou chybu
- D** cyklus nemusí proběhnout ani jednou

**5** Po vykonání následujícího kódu bude v `buf` uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```

- A** "Hello World! , Hello World!!"
- B** "Hello World!"
- C** "Hello, Hello World!!"
- D** "Hello World! , Hello World!"
- E** kód se nepřeloží (obsahuje chybu)
- F** "!"

**6** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A** 6
- B** kód se nepřeloží (obsahuje chybu)
- C** 5
- D** 9
- E** 3

**7** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = **pA + 200;
Bude obsah proměně A roven:
```

- A** 100
- B** kód obsahuje syntaktickou chybu
- C** 200
- D** 300
- E** obsah hodnoty A nelze bez spuštění jednoznačně určit

**8** Po provedení kódu:

```
int A = 100;
A = --A + 200;
Bude obsah proměně A roven:
```

- A** 300
- B** 101
- C** 100
- D** 200
- E** 299
- F** kód obsahuje syntaktickou chybu

**9** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A** Pole o 6 znacích a textový řetězec o délce 6 znaků
- B** Pole o 5 znacích a textový řetězec o délce 6 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 6 znacích a textový řetězec o délce 5 znaků
- E** Pole o 5 znacích a textový řetězec o délce 5 znaků

- 10** Po provedení kódu:  

```
int A = 100;  
switch (A) {  
    case 100: A++;  
    case 101: A++; break;  
    case 102: A++;  
}
```

bude obsah proměnné A roven:
- A** 102
  - B** 103
  - C** kód obsahuje syntaktickou chybu
  - D** 101
- 
- 11** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x += 1; }
```
- A** 0
  - B** 99
  - C** 1
  - D** kód se nepřeloží (obsahuje chybu)
  - E** 100
- 
- 12** Hodnota výrazu  $(2+3)/5+5$  je:
- A** 0
  - B** 2
  - C** 10
  - D** 1
  - E** 6
- 
- 13** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
}
```

proběhne:
- A** 100x
  - B** kód se nepřeloží (obsahuje chybu)
  - C** 99x
  - D** 0x
  - E** 101x
- 
- 14** Výsledkem výrazu  $(1 \ \&\& \ 1)$  je:
- A** není definováno
  - B** 10
  - C** 0
  - D** 1

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		124

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:
- A 0
  - B není definováno
  - C 0 pokud je A rovno 0
  - D 1

- 2** Po provedení kódu:
- ```
int A = 100;
int* pA = &A;
A = **pA + 200;
```
- Bude obsah proměnné A roven:
- A 100
  - B 200
  - C kód obsahuje syntaktickou chybu
  - D 300
  - E obsah hodnoty A nelze bez spuštění jednoznačně určit

- 3** Tělo cyklu
- ```
for (int i = 1; i < 100; i++) {
}
```
- proběhne:
- A 0x
  - B kód se nepřeloží (obsahuje chybu)
  - C 100x
  - D 99x
  - E 101x

- 4** Cyklus
- ```
int i = 0;
while (i <= 100) {
    i++;
}
```
- proběhne:
- A 100x
  - B 101x
  - C 99x
  - D kód obsahuje syntaktickou chybu

- 5** Následující zápis reprezentuje:
- ```
int x[10];
```
- A Pole s 9 prvky s indexy 0..9 typu int
  - B Pole s 9 prvky s indexy 1..9 typu int
  - C Pole s 10 prvky s indexy 0..9 typu int
  - D kód se nepřeloží (obsahuje chybu)
  - E Pole s 10 prvky s indexy 1..10 typu int

- 6** Po provedení kódu:
- ```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```
- bude obsah proměnné A roven:
- A 102
  - B 101
  - C kód obsahuje syntaktickou chybu
  - D 100

- 7** Následující kód reprezentuje:
- ```
char text[] = "Nazdar.";
```
- A kód se nepřeloží (obsahuje chybu)
  - B Pole o 7 znacích a textový řetězec o délce 8 znaků
  - C Pole o 8 znacích a textový řetězec o délce 7 znaků
  - D Pole o 7 znacích a textový řetězec o délce 7 znaků
  - E Pole o 8 znacích a textový řetězec o délce 8 znaků

- 8** Po vykonání následujícího kódu bude v buf uložen řetězec:
- ```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcat(buf, " , Hello World!");
strcat(buf, "!");
```
- A kód se nepřeloží (obsahuje chybu)
  - B "!"
  - C "Hello, Hello World!!"
  - D "Hello World! , Hello World!"
  - E "Hello World! , Hello World!!"
  - F "Hello World!"

- 9** Hodnota proměnné po provedení následujícího kódu bude:
- ```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```
- A 3
  - B kód se nepřeloží (obsahuje chybu)
  - C 5
  - D 6
  - E 9

**10** Cyklus  
`int i = 100;`  
`do {`  
    `i++;`  
`} while (i < 100);`  
proběhne:

- A** jednou
- B** kód obsahuje syntaktickou chybu
- C** ani jednou
- D** dvakrát

**11** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A** 40
- B** 30
- C** 60
- D** 0
- E** 4

**12** Datový typ `int` je:

- A** Typ bez znaménka pro uchovávání celých čísel
- B** Typ se znaménkem pro uchovávání celých čísel
- C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou

**13** Hodnota proměnné `x` po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A** 0
- B** 1
- C** 101
- D** -1
- E** kód se nepřeloží (obsahuje chybu)

**14** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
```

Bude obsah proměnné `A` roven:

- A** 101
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 300
- E** 200
- F** 299



# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		125

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Po provedení kódu:  
`int A = 100;  
A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 101
  - B** 200
  - C** 300
  - D** 301
  - E** kód obsahuje syntaktickou chybu
  - F** 100

- 2** Cyklus  
`int i = 100;  
do {  
    i++;  
} while (i < 100);`  
proběhne:
- A** ani jednou
  - B** jednou
  - C** dvakrát
  - D** kód obsahuje syntaktickou chybu

- 3** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};`
- A** 0
  - B** -1
  - C** 1
  - D** kód se nepřełoży (obsahuje chybu)
  - E** 101

- 4** Výsledkem výrazu `(1 && 1)` je:
- A** není definováno
  - B** 1
  - C** 0
  - D** 10

- 5** Cyklus  
`int i = 0;  
while (i < 100) {  
    i++;  
}`  
proběhne:
- A** 101x
  - B** 99x
  - C** 100x
  - D** kód obsahuje syntaktickou chybu

- 6** Datový typ `int` má vždy velikost:
- A** 8 bajty
  - B** 4 bajty
  - C** 2 bajty
  - D** v závislosti na architektuře

- 7** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** kód se nepřełoży (obsahuje chybu)
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - D** Pole o 6 znacích a textový řetězec o délce 6 znaků
  - E** Pole o 5 znacích a textový řetězec o délce 6 znaků

- 8** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 0
  - B** 44
  - C** 1
  - D** 33
  - E** 66

- 9** Následující zápis reprezentuje:  
`int x[10];`
- A** Pole s 10 prvky s indexy 0..9 typu `int`
  - B** kód se nepřełoży (obsahuje chybu)
  - C** Pole s 9 prvky s indexy 1..9 typu `int`
  - D** Pole s 9 prvky s indexy 0..9 typu `int`
  - E** Pole s 10 prvky s indexy 1..10 typu `int`

- 10** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {  
    }  
}`  
proběhne:
- A** 99x
  - B** 101x
  - C** 0x
  - D** 100x
  - E** kód se nepřełoży (obsahuje chybu)

- 11** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];`
- A** 6
  - B** 9
  - C** 3
  - D** kód se nepřełoży (obsahuje chybu)
  - E** 5

- 12** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = *A + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** kód obsahuje syntaktickou chybu
  - C** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - D** 200
  - E** 100

- 13** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};`  
`strcpy(buf, "Hello World!");`  
`strcat(buf, " , Hello World!");`  
`strcat(buf, "!");`
- A** "!"
  - B** "Hello World!"
  - C** "Hello World! , Hello World!!"
  - D** kód se nepřeloží (obsahuje chybu)
  - E** "Hello, Hello World!!"
  - F** "Hello World! , Hello World!"

- 14** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
    `default: A--;`  
`}`  
bude obsah proměnné A roven:
- A** 98
  - B** 100
  - C** 97
  - D** 101
  - E** kód obsahuje syntaktickou chybu

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		126

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:

**A** alespoň jednou  
**B** kód obsahuje syntaktickou chybu  
**C** vždy 100x  
**D** cyklus nemusí proběhnout ani jednou
- 2** Výsledkem výrazu (1 && 1) je:

**A** 1  
**B** není definováno  
**C** 10  
**D** 0
- 3** Po provedení kódu:  
int A = 100;  
int\* pA = &A;  
A = pA + 200;  
Bude obsah proměné A roven:

**A** 200  
**B** 300  
**C** kód obsahuje syntaktickou chybu  
**D** 100  
**E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 4** Po vykonání následujícího kódu bude v buf uložen řetězec:  
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!!");

**A** "Hello World! , Hello World!!"  
**B** kód se nepřeloží (obsahuje chybu)  
**C** "Hello World! , Hello World!"  
**D** "Hello World!"  
**E** "!"  
**F** "Hello, Hello World!!"
- 5** Tělo cyklu  
for (int i = 1; i <= 100; i++) {  
}  
proběhne:

**A** kód se nepřeloží (obsahuje chybu)  
**B** 100x  
**C** 101x  
**D** 0x  
**E** 99x
- 6** Datový typ int je:

**A** Typ bez znaménka pro uchovávání celých čísel  
**B** Typ se znaménkem pro uchovávání celých čísel  
**C** Typ bez znaménka pro uchovávání čísel s desetinou čárkou  
**D** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- 7** Hodnota výrazu (2/3)\*(10+11+12)\*2 je:

**A** 66  
**B** 33  
**C** 0  
**D** 1  
**E** 44
- 8** Následující zápis reprezentuje:  
int x[10];

**A** Pole s 10 prvky s indexy 0..9 typu int  
**B** Pole s 10 prvky s indexy 1..10 typu int  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 9 prvky s indexy 1..9 typu int  
**E** Pole s 9 prvky s indexy 0..9 typu int
- 9** Hodnota proměnné po provedení následujícího kódu bude:  
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];

**A** 3  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 5  
**D** 9  
**E** 6
- 10** Po provedení kódu:  
int A = 100;  
A = A-- + 200;  
Bude obsah proměné A roven:

**A** 299  
**B** kód obsahuje syntaktickou chybu  
**C** 100  
**D** 101  
**E** 200  
**F** 300

**11** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++);
{ x += 1; }
```

- A 1
- B kód se nepřeloží (obsahuje chybu)
- C 99
- D 0
- E 100

**12** Cyklus

```
while (i < 100) {
    i++;
}
```

kde i je proměnná typu int proběhne:

- A vždy 100x
- B alespoň jednou
- C kód obsahuje syntaktickou chybu
- D cyklus nemusí proběhnout ani jednou

**13** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměnné A roven:

- A 98
- B kód obsahuje syntaktickou chybu
- C 102
- D 101
- E 100

**14** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 6 znacích a textový řetězec o délce 6 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 5 znacích a textový řetězec o délce 5 znaků
- D Pole o 6 znacích a textový řetězec o délce 5 znaků
- E Pole o 5 znacích a textový řetězec o délce 6 znaků

# PB071: Prubezny test pb071 13:00

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		127

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Následující zápis reprezentuje:  
`double x[10];`

- A** Pole s 10 prvky s indexy 0..9 typu double
- B** Pole s 10 prvky s indexy 1..10 typu double
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole s 9 prvky s indexy 1..9 typu double
- E** Pole s 9 prvky s indexy 0..9 typu double

**2** Po provedení kódu:

```
int A = 100;
```

```
A = --A + 200;
```

Bude obsah proměnné A roven:

- A** 200
- B** 100
- C** 300
- D** 101
- E** kód obsahuje syntaktickou chybu
- F** 299

**3** Po provedení kódu:

```
int A = 100;
```

```
int* pA = &A;
```

```
A = **pA + 200;
```

Bude obsah proměnné A roven:

- A** kód obsahuje syntaktickou chybu
- B** 100
- C** 200
- D** obsah hodnoty A nelze bez spuštění jednoznačně určit
- E** 300

**4** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {  
}
```

proběhne:

- A** 0x
- B** kód se nepřeloží (obsahuje chybu)
- C** 101x
- D** 100x
- E** 99x

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
```

```
int y = 0;
```

```
for (int i = 1; i < 3; i++)
```

```
    y = y + x[i];
```

- A** 3
- B** 9
- C** kód se nepřeloží (obsahuje chybu)
- D** 5
- E** 6

**6** Cyklus

```
while (i < 100) {
```

```
    i++;
```

```
}
```

kde i je proměnná typu int proběhne:

- A** alespoň jednou
- B** kód obsahuje syntaktickou chybu
- C** vždy 100x
- D** cyklus nemusí proběhnout ani jednou

**7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
```

```
strcpy(buf, "Hello World!");
```

```
strcat(buf, " , Hello World!");
```

```
strcat(buf, "!");
```

- A** "Hello World! , Hello World!"
- B** "Hello World!"
- C** "Hello, Hello World!!"
- D** "!"
- E** "Hello World! , Hello World!!"
- F** kód se nepřeloží (obsahuje chybu)

**8** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A** 66
- B** 33
- C** 1
- D** 0
- E** 44

**9** Po provedení kódu:

```
int A = 98;
```

```
switch (A) {
```

```
    case 99: A++; break;
```

```
    case 100: A++; break;
```

```
    case 101: A++; break;
```

```
}
```

bude obsah proměnné A roven:

- A** 102
- B** 100
- C** 101
- D** 98
- E** kód obsahuje syntaktickou chybu

- 10** Hodnota proměnné x po provedení následující kódu bude:
- ```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```
- A** -1  
**B** 101  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 0  
**E** 1
- 
- 11** Cyklus
- ```
do {
    i++;
} while (i < 100);
```
- kde i je proměnná typu int proběhne:
- A** kód obsahuje syntaktickou chybu  
**B** cyklus nemusí proběhnout ani jednou  
**C** vždy 100x  
**D** alespoň jednou
- 
- 12** Výsledkem výrazu (1 && 0) je:
- A** není definováno  
**B** 10  
**C** 1  
**D** 0
- 
- 13** Datový typ int má vždy velikost:
- A** 2 bajty  
**B** 4 bajty  
**C** 8 bajty  
**D** v závislosti na architektuře
- 
- 14** Následující kód reprezentuje:
- ```
char text[] = "Ahoj!";
```
- A** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**B** Pole o 6 znacích a textový řetězec o délce 5 znaků  
**C** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**E** kód se nepřeloží (obsahuje chybu)

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 128          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Následující zápis reprezentuje:  
`double x[10];`
- A** kód se nepřeloží (obsahuje chybu)  
**B** Pole s 10 prvků s indexy 0..9 typu double  
**C** Pole s 9 prvků s indexy 0..9 typu double  
**D** Pole s 9 prvků s indexy 1..9 typu double  
**E** Pole s 10 prvků s indexy 1..10 typu double

- 2** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");`
- A** "Hello World! , Hello World!"  
**B** "Hello, Hello World!!"  
**C** kód se nepřeloží (obsahuje chybu)  
**D** "!"  
**E** "Hello World!"  
**F** "Hello World! , Hello World!!"

- 3** Po provedení kódu:  
`int A = 100;  
A = ++A + 200;`  
Bude obsah proměnné A roven:
- A** 200  
**B** kód obsahuje syntaktickou chybu  
**C** 100  
**D** 301  
**E** 101  
**F** 300

- 4** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměnná typu integer je:
- A** 0  
**B** není definováno  
**C** 0 pokud je A rovno 0  
**D** 1

- 5** Cyklus  
`do {  
    i++;  
} while (i < 100);`  
kde i je proměnná typu int proběhne:
- A** cyklus nemusí proběhnout ani jednou  
**B** kód obsahuje syntaktickou chybu  
**C** alespoň jednou  
**D** vždy 100x

- 6** Po provedení kódu:  
`int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}`  
bude obsah proměnné A roven:
- A** kód obsahuje syntaktickou chybu  
**B** 102  
**C** 100  
**D** 101

- 7** Hodnota výrazu  $(2-20)/(5+5)$  je:
- A** -1.8  
**B** 2  
**C** 0  
**D** -1  
**E** 1

- 8** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;  
for (int i = 0; i < 100; i++)  
{ x -= 1; }`
- A** -100  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 1  
**D** -99  
**E** 0

- 9** Tělo cyklu  
`for (int i = 0; i < 100; i++) {  
}`  
proběhne:
- A** kód se nepřeloží (obsahuje chybu)  
**B** 99x  
**C** 0x  
**D** 101x  
**E** 100x

- 10** Hodnota proměnné po provedení následujícího kódu bude:  
`int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];`
- A** 3  
**B** 5  
**C** 9  
**D** kód se nepřeloží (obsahuje chybu)  
**E** 6

- 11** Cyklus  
`while (i < 100) {  
 i++;  
}`  
kde `i` je proměnná typu `int` proběhne:
- A** vždy 100x
  - B** alespoň jednou
  - C** kód obsahuje syntaktickou chybu
  - D** cyklus nemusí proběhnout ani jednou
- 
- 12** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - B** Pole o 7 znacích a textový řetězec o délce 8 znaků
  - C** Pole o 8 znacích a textový řetězec o délce 7 znaků
  - D** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = **pA + 200;`  
Bude obsah proměnné `A` roven:
- A** obsah hodnoty `A` nelze bez spuštění jednoznačně určit
  - B** 300
  - C** 200
  - D** kód obsahuje syntaktickou chybu
  - E** 100
- 
- 14** Datový typ `char` je:
- A** Typ pro uchovávání malých celých čísel nebo znaků
  - B** Typ pro uchovávání velkých celých čísel
  - C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - D** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
-



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 129          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 100;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
}
```

bude obsah proměně A roven:

- A 102
- B 101
- C kód obsahuje syntaktickou chybu
- D 100

**2** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
for (int i = 0; i < 100; i++) {
    x += 1; }
```

- A 1
- B 99
- C kód se nepřeloží (obsahuje chybu)
- D 100
- E 0

**3** Cyklus

```
int i = 99;
do {
    i++;
} while (i < 100);
```

proběhne:

- A kód obsahuje syntaktickou chybu
- B jednou
- C ani jednou
- D dvakrát

**4** Následující kód reprezentuje:

```
char text[] = "Ahoj!";
```

- A Pole o 5 znacích a textový řetězec o délce 5 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 5 znacích a textový řetězec o délce 6 znaků
- D Pole o 6 znacích a textový řetězec o délce 5 znaků
- E Pole o 6 znacích a textový řetězec o délce 6 znaků

**5** Hodnota výrazu  $(3/2) * (10+10) * 2$  je:

- A 4
- B 0
- C 60
- D 40
- E 30

**6** Výsledkem výrazu  $(1 \ \&\& \ 0)$  je:

- A 1
- B 10
- C není definováno
- D 0

**7** Tělo cyklu

```
for (int i = 0; i < 100; i++) {
}
```

proběhne:

- A 0x
- B 99x
- C 101x
- D kód se nepřeloží (obsahuje chybu)
- E 100x

**8** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i <= 3; i++)
    y = y + x[i];
```

- A 3
- B 5
- C 6
- D 9
- E kód se nepřeloží (obsahuje chybu)

**9** Po provedení kódu:

```
int A = 100;
int* pA = &A;
A = *A + 200;
```

Bude obsah proměně A roven:

- A 100
- B 300
- C obsah hodnoty A nelze bez spuštění jednoznačně určit
- D kód obsahuje syntaktickou chybu
- E 200

**10** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};
strcpy(buf, "Hello World!");
strcpy(buf, " , Hello World!");
strcat(buf, "!!");
```

- A "Hello World!"
- B kód se nepřeloží (obsahuje chybu)
- C " , Hello World!!"
- D "!"
- E "Hello, Hello World!!"
- F "Hello World! , Hello World!"

- 11** Datový typ char je:
- A** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
  - B** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
  - C** Typ pro uchovávání malých celých čísel nebo znaků
  - D** Typ pro uchovávání velkých celých čísel

- 12** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:
- A** 100
  - B** 101
  - C** 301
  - D** 200
  - E** kód obsahuje syntaktickou chybu
  - F** 300

- 13** Cyklus  
`int i = 0;`  
`while (i < 100) {`  
    `i++;`  
`}`  
proběhne:
- A** 100x
  - B** kód obsahuje syntaktickou chybu
  - C** 101x
  - D** 99x

- 14** Následující zápis reprezentuje:  
`int x[10];`
- A** Pole s 10 prvky s indexy 1..10 typu int
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole s 10 prvky s indexy 0..9 typu int
  - D** Pole s 9 prvky s indexy 1..9 typu int
  - E** Pole s 9 prvky s indexy 0..9 typu int

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 130          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po vykonání následujícího kódu bude v buf uložen řetězec:  

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello World! , Hello World!!"
- B "Hello World! , Hello World!"
- C "Hello, Hello World!!"
- D kód se nepřeloží (obsahuje chybu)
- E "Hello World!"
- F "!"

**2** Datový typ char je:

- A Typ pro uchovávání malých celých čísel nebo znaků
- B Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- C Typ pro uchovávání velkých celých čísel
- D Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**3** Cyklus  

```
int i = 0;  
while (i <= 100) {  
    i++;  
}  
proběhne:
```

- A 101x
- B 99x
- C kód obsahuje syntaktickou chybu
- D 100x

**4** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:

- A 44
- B 33
- C 0
- D 66
- E 1

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 3
- C 5
- D 9
- E 6

**6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};
```

- A 0
- B 1
- C -1
- D kód se nepřeloží (obsahuje chybu)
- E 101

**7** Cyklus  

```
int i = 100;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A dvakrát
- B jednou
- C kód obsahuje syntaktickou chybu
- D ani jednou

**8** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
}  
proběhne:
```

- A 100x
- B 0x
- C 99x
- D 101x
- E kód se nepřeloží (obsahuje chybu)

**9** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;  
Bude obsah proměnné A roven:
```

- A kód obsahuje syntaktickou chybu
- B obsah hodnoty A nelze bez spuštění jednoznačně určit
- C 300
- D 200
- E 100

**10** Po provedení kódu:  
`int A = 100;`  
`A = A++ + 200;`  
Bude obsah proměnné A roven:  
**A** 200  
**B** 100  
**C** 101  
**D** kód obsahuje syntaktickou chybu  
**E** 301  
**F** 300

**11** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`  
**A** Pole o 6 znacích a textový řetězec o délce 6 znaků  
**B** kód se nepřeloží (obsahuje chybu)  
**C** Pole o 5 znacích a textový řetězec o délce 6 znaků  
**D** Pole o 5 znacích a textový řetězec o délce 5 znaků  
**E** Pole o 6 znacích a textový řetězec o délce 5 znaků

**12** Po provedení kódu:  
`int A = 98;`  
`switch (A) {`  
    `case 99: A++; break;`  
    `case 100: A++; break;`  
    `case 101: A++; break;`  
    `default: A--;`  
`}`  
bude obsah proměnné A roven:  
**A** 101  
**B** kód obsahuje syntaktickou chybu  
**C** 98  
**D** 97  
**E** 100

**13** Následující zápis reprezentuje:  
`double x[10];`  
**A** Pole s 9 prvky s indexy 0..9 typu double  
**B** Pole s 9 prvky s indexy 1..9 typu double  
**C** kód se nepřeloží (obsahuje chybu)  
**D** Pole s 10 prvky s indexy 0..9 typu double  
**E** Pole s 10 prvky s indexy 1..10 typu double

**14** Výsledkem výrazu `(1 && 0)` je:  
**A** 10  
**B** 0  
**C** není definováno  
**D** 1

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 131          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 100;  
A = A-- + 200;  
Bude obsah proměné A roven:
```

- A** 299
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 300
- E** 101
- F** 200

**2** Cyklus

```
int i = 0;  
while (i < 100) {  
    i++;  
}  
proběhne:
```

- A** 100x
- B** kód obsahuje syntaktickou chybu
- C** 101x
- D** 99x

**3** Tělo cyklu

```
for (int i = 1; i <= 100; i++) {  
}  
proběhne:
```

- A** 99x
- B** 101x
- C** 100x
- D** 0x
- E** kód se nepřeloží (obsahuje chybu)

**4** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i < 3; i++)  
    y = y + x[i];
```

- A** kód se nepřeloží (obsahuje chybu)
- B** 9
- C** 6
- D** 5
- E** 3

**5** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** "Hello World! , Hello World!"
- B** "!"
- C** " , Hello World!!"
- D** "Hello, Hello World!!"
- E** kód se nepřeloží (obsahuje chybu)
- F** "Hello World!"

**6** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;  
for (int i = 0; i < 100; i++)  
{ x += 1; }
```

- A** 99
- B** kód se nepřeloží (obsahuje chybu)
- C** 100
- D** 1
- E** 0

**7** Cyklus

```
int i = 0;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A** 100x
- B** 99x
- C** 101x
- D** kód obsahuje syntaktickou chybu

**8** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 8 znacích a textový řetězec o délce 8 znaků
- B** Pole o 7 znacích a textový řetězec o délce 8 znaků
- C** Pole o 7 znacích a textový řetězec o délce 7 znaků
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole o 8 znacích a textový řetězec o délce 7 znaků

**9** Datový typ char je:

- A** Typ bez znaménka pro uchovávání čísel s desetinou čárkou
- B** Typ pro uchovávání velkých celých čísel
- C** Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D** Typ pro uchovávání malých celých čísel nebo znaků

- 10** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:
- A** 101
  - B** 102
  - C** kód obsahuje syntaktickou chybu
  - D** 100
  - E** 98
- 
- 11** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** není definováno
  - B** 0
  - C** 0 pokud je A rovno 0
  - D** nenulová hodnota
- 
- 12** Po provedení kódu:  

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;
```

Bude obsah proměnné A roven:
- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - B** 100
  - C** 300
  - D** 200
  - E** kód obsahuje syntaktickou chybu
- 
- 13** Následující zápis reprezentuje:  

```
double x[10];
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** Pole s 10 prvky s indexy 1..10 typu double
  - C** Pole s 9 prvky s indexy 1..9 typu double
  - D** Pole s 10 prvky s indexy 0..9 typu double
  - E** Pole s 9 prvky s indexy 0..9 typu double
- 
- 14** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:
- A** 1
  - B** 66
  - C** 0
  - D** 33
  - E** 44

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 132          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Po provedení kódu:

```
int A = 100;  
A = A++ + 200;  
Bude obsah proměné A roven:
```

- A** 300
- B** kód obsahuje syntaktickou chybu
- C** 100
- D** 301
- E** 101
- F** 200

**2** Následující zápis reprezentuje:

```
double x[10];
```

- A** Pole s 10 prvky s indexy 1..10 typu double
- B** Pole s 9 prvky s indexy 1..9 typu double
- C** Pole s 10 prvky s indexy 0..9 typu double
- D** kód se nepřeloží (obsahuje chybu)
- E** Pole s 9 prvky s indexy 0..9 typu double

**3** Cyklus

```
int i = 0;  
do {  
    i++;  
} while (i < 100);  
proběhne:
```

- A** 99x
- B** 100x
- C** kód obsahuje syntaktickou chybu
- D** 101x

**4** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");
```

- A** kód se nepřeloží (obsahuje chybu)
- B** "Hello World! , Hello World!"
- C** "Hello World!"
- D** "Hello, Hello World!!"
- E** "Hello World! , Hello World!!"
- F** "!"

**5** Následující kód reprezentuje:

```
char text[] = "Nazdar.";
```

- A** Pole o 7 znacích a textový řetězec o délce 8 znaků
- B** Pole o 8 znacích a textový řetězec o délce 8 znaků
- C** kód se nepřeloží (obsahuje chybu)
- D** Pole o 8 znacích a textový řetězec o délce 7 znaků
- E** Pole o 7 znacích a textový řetězec o délce 7 znaků

**6** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A** 9
- B** 6
- C** 5
- D** 3
- E** kód se nepřeloží (obsahuje chybu)

**7** Po provedení kódu:

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměné A roven:

- A** 98
- B** kód obsahuje syntaktickou chybu
- C** 101
- D** 100
- E** 102

**8** Hodnota výrazu  $(2-20)/(5+5)$  je:

- A** 2
- B** -1
- C** 1
- D** -1.8
- E** 0

**9** Datový typ int má vždy velikost:

- A** 2 bajty
- B** 8 bajty
- C** v závislosti na architektuře
- D** 4 bajty

**10** Po provedení kódu:

```
int A = 100;  
int* pA = &A;  
A = &pA + 200;  
Bude obsah proměné A roven:
```

- A** obsah hodnoty A nelze bez spuštění jednoznačně určit
- B** kód obsahuje syntaktickou chybu
- C** 200
- D** 100
- E** 300

- 11** Výsledkem výrazu  $(1 \parallel A)$ , kde A je proměnná typu integer je:
- A** není definováno
  - B** 0
  - C** 0 pokud je A rovno 0
  - D** nenulová hodnota
- 
- 12** Tělo cyklu  

```
for (int i = 1; i <= 100; i++) {  
    }  
proběhne:
```
- A** kód se nepřeloží (obsahuje chybu)
  - B** 0x
  - C** 100x
  - D** 99x
  - E** 101x
- 
- 13** Cyklus  

```
while (i < 100) {  
    i++;  
}  
kde i je proměnná typu int proběhne:
```
- A** cyklus nemusí proběhnout ani jednou
  - B** alespoň jednou
  - C** kód obsahuje syntaktickou chybu
  - D** vždy 100x
- 
- 14** Hodnota proměnné x po provedení následující kódu bude:  

```
int x = 0;  
for (int i = 0; i < 100; i++);  
{ x -= 1; }
```
- A** -100
  - B** 1
  - C** 0
  - D** -99
  - E** kód se nepřeloží (obsahuje chybu)



# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 133          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota proměnné po provedení následujícího kódu bude:  

```
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];
```

- A kód se nepřeloží (obsahuje chybu)
- B 5
- C 6
- D 9
- E 3

**2** Datový typ char je:

- A Typ pro uchovávání velkých celých čísel
- B Typ pro uchovávání malých celých čísel nebo znaků
- C Typ se znaménkem pro uchovávání čísel s desetinou čárkou
- D Typ bez znaménka pro uchovávání čísel s desetinou čárkou

**3** Cyklus  

```
int i = 100;  
do {  
    i++;  
} while (i < 100);
```

proběhne:

- A jednou
- B dvakrát
- C ani jednou
- D kód obsahuje syntaktickou chybu

**4** Po provedení kódu:  

```
int A = 98;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}
```

bude obsah proměnné A roven:

- A 98
- B kód obsahuje syntaktickou chybu
- C 100
- D 102
- E 101

**5** Výsledkem výrazu (1 && 0) je:

- A 10
- B 1
- C není definováno
- D 0

**6** Následující zápis reprezentuje:  

```
int x[10];
```

- A Pole s 10 prvky s indexy 0..9 typu int
- B Pole s 9 prvky s indexy 0..9 typu int
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 10 prvky s indexy 1..10 typu int
- E Pole s 9 prvky s indexy 1..9 typu int

**7** Po vykonání následujícího kódu bude v buf uložen řetězec:

```
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcpy(buf, " , Hello World!");  
strcat(buf, "!");
```

- A "Hello, Hello World!!"
- B "Hello World! , Hello World!"
- C "!"
- D "Hello World!"
- E " , Hello World!!"
- F kód se nepřeloží (obsahuje chybu)

**8** Tělo cyklu  

```
for (int i = 0; i < 100; i++) {  
    }  
proběhne:
```

- A 0x
- B 99x
- C kód se nepřeloží (obsahuje chybu)
- D 100x
- E 101x

**9** Následující kód reprezentuje:  

```
char text[] = "Nazdar.";
```

- A Pole o 7 znacích a textový řetězec o délce 7 znaků
- B kód se nepřeloží (obsahuje chybu)
- C Pole o 8 znacích a textový řetězec o délce 8 znaků
- D Pole o 8 znacích a textový řetězec o délce 7 znaků
- E Pole o 7 znacích a textový řetězec o délce 8 znaků

**10** Cyklus  

```
while (i < 100) {  
    i++;  
}
```

kde i je proměnná typu int proběhne:

- A vždy 100x
- B cyklus nemusí proběhnout ani jednou
- C alespoň jednou
- D kód obsahuje syntaktickou chybu

- 11** Po provedení kódu:  
`int A = 100;`  
`int* pA = &A;`  
`A = pA + 200;`  
Bude obsah proměnné A roven:  
**A** kód obsahuje syntaktickou chybu  
**B** 200  
**C** 300  
**D** obsah hodnoty A nelze bez spuštění jednoznačně určit  
**E** 100
- 
- 12** Hodnota výrazu  $(2/3) * (10+11+12) * 2$  je:  
**A** 1  
**B** 44  
**C** 66  
**D** 33  
**E** 0
- 
- 13** Po provedení kódu:  
`int A = 100;`  
`A = ++A + 200;`  
Bude obsah proměnné A roven:  
**A** 100  
**B** kód obsahuje syntaktickou chybu  
**C** 301  
**D** 200  
**E** 101  
**F** 300
- 
- 14** Hodnota proměnné x po provedení následující kódu bude:  
`int x = 0;`  
`for (int i = 0; i < 100; i++);`  
`{ x += 1; }`  
**A** kód se nepřeloží (obsahuje chybu)  
**B** 1  
**C** 99  
**D** 0  
**E** 100

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení – pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 134          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

**1** Hodnota výrazu  $(2/3) \cdot (10+11+12) \cdot 2$  je:

- A 44
- B 33
- C 66
- D 0
- E 1

**2** Cyklus  

```
int i = 0;
while (i < 100) {
    i++;
}
```

proběhne:

- A 100x
- B 101x
- C 99x
- D kód obsahuje syntaktickou chybu

**3** Hodnota proměnné x po provedení následující kódu bude:

```
int x = 0;
if (1)
    x = x + 1;
else
    x = x - 1;
for (int i = 0; i < 100; i++) {x++};
```

- A -1
- B 0
- C kód se nepřeloží (obsahuje chybu)
- D 1
- E 101

**4** Po provedení kódu:

```
int A = 100;
A = A-- + 200;
Bude obsah proměně A roven:
```

- A 101
- B kód obsahuje syntaktickou chybu
- C 299
- D 300
- E 100
- F 200

**5** Hodnota proměnné po provedení následujícího kódu bude:

```
int x[] = { 1, 2, 3, 4, 5, 0 };
int y = 0;
for (int i = 1; i < 3; i++)
    y = y + x[i];
```

- A 9
- B kód se nepřeloží (obsahuje chybu)
- C 3
- D 5
- E 6

**6** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:

- A 0
- B není definováno
- C 0 pokud je A rovno 0
- D 1

**7** Po provedení kódu:

```
int A = 98;
switch (A) {
    case 99: A++; break;
    case 100: A++; break;
    case 101: A++; break;
    default: A--;
}
```

bude obsah proměně A roven:

- A 101
- B 100
- C 98
- D kód obsahuje syntaktickou chybu
- E 97

**8** Následující zápis reprezentuje:

```
int x[10];
```

- A Pole s 9 prvky s indexy 1..9 typu int
- B Pole s 10 prvky s indexy 0..9 typu int
- C kód se nepřeloží (obsahuje chybu)
- D Pole s 10 prvky s indexy 1..10 typu int
- E Pole s 9 prvky s indexy 0..9 typu int

**9** Datový typ int má vždy velikost:

- A v závislosti na architektuře
- B 8 bajty
- C 4 bajty
- D 2 bajty

**10** Cyklus

```
do {
    i++;
} while (i < 100);
```

kde i je proměná typu int proběhne:

- A cyklus nemusí proběhnout ani jednou
- B kód obsahuje syntaktickou chybu
- C vždy 100x
- D alespoň jednou

- 11** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = pA + 200;`  
Bude obsah proměnné A roven:
- A** 300
  - B** 200
  - C** 100
  - D** obsah hodnoty A nelze bez spuštění jednoznačně určit
  - E** kód obsahuje syntaktickou chybu
- 
- 12** Tělo cyklu  
`for (int i = 0; i <= 100; i++) {  
}`  
proběhne:
- A** 0x
  - B** 100x
  - C** 101x
  - D** 99x
  - E** kód se nepřeloží (obsahuje chybu)
- 
- 13** Následující kód reprezentuje:  
`char text[] = "Nazdar.";`
- A** Pole o 7 znacích a textový řetězec o délce 7 znaků
  - B** kód se nepřeloží (obsahuje chybu)
  - C** Pole o 8 znacích a textový řetězec o délce 8 znaků
  - D** Pole o 8 znacích a textový řetězec o délce 7 znaků
  - E** Pole o 7 znacích a textový řetězec o délce 8 znaků
- 
- 14** Po vykonání následujícího kódu bude v buf uložen řetězec:  
`char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");`
- A** kód se nepřeloží (obsahuje chybu)
  - B** "Hello World! , Hello World!!"
  - C** "Hello, Hello World!!"
  - D** "!"
  - E** "Hello World! , Hello World!"
  - F** "Hello World!"

# PB071: Prubezny test pb071 13:00

| Jméno a příjmení - pište do okénka | UČO | Číslo zadání |
|------------------------------------|-----|--------------|
|                                    |     | 135          |

Vždy prave jedna odpoved je spravna. Otazky jsou bodovany 1, 2 nebo 3 body. Za spatnou odpoved mohou byt strzeny body.

- 1** Cyklus  
do {  
    i++;  
} while (i < 100);  
kde i je proměná typu int proběhne:  
**A** vždy 100x  
**B** cyklus nemusí proběhnout ani jednou  
**C** alespoň jednou  
**D** kód obsahuje syntaktickou chybu

- 2** Po provedení kódu:  
int A = 100;  
A = A-- + 200;  
Bude obsah proměné A roven:  
**A** 100  
**B** 300  
**C** 299  
**D** 101  
**E** kód obsahuje syntaktickou chybu  
**F** 200

- 3** Následující zápis reprezentuje:  
int x[10];  
**A** Pole s 9 prvky s indexy 0..9 typu int  
**B** Pole s 10 prvky s indexy 0..9 typu int  
**C** Pole s 10 prvky s indexy 1..10 typu int  
**D** Pole s 9 prvky s indexy 1..9 typu int  
**E** kód se nepřeloží (obsahuje chybu)

- 4** Hodnota proměnné po provedení následujícího kódu bude:  
int x[] = { 1, 2, 3, 4, 5, 0 };  
int y = 0;  
for (int i = 1; i <= 3; i++)  
    y = y + x[i];  
**A** 5  
**B** 9  
**C** kód se nepřeloží (obsahuje chybu)  
**D** 3  
**E** 6

- 5** Tělo cyklu  
for (int i = 0; i < 100; i++) {  
}  
proběhne:  
**A** 99x  
**B** kód se nepřeloží (obsahuje chybu)  
**C** 101x  
**D** 0x  
**E** 100x

- 6** Hodnota výrazu  $(3/2)*(10+10)*2$  je:  
**A** 4  
**B** 40  
**C** 0  
**D** 30  
**E** 60

- 7** Po provedení kódu:  
int A = 100;  
switch (A) {  
    case 99: A++; break;  
    case 100: A++; break;  
    case 101: A++; break;  
}  
bude obsah proměné A roven:  
**A** 100  
**B** 102  
**C** kód obsahuje syntaktickou chybu  
**D** 101

- 8** Po vykonání následujícího kódu bude v buf uložen řetězec:  
char buf[1024] = {0};  
strcpy(buf, "Hello World!");  
strcat(buf, " , Hello World!");  
strcat(buf, "!");  
**A** kód se nepřeloží (obsahuje chybu)  
**B** "Hello, Hello World!!"  
**C** "Hello World! , Hello World!!"  
**D** "Hello World!"  
**E** "Hello World! , Hello World!"  
**F** "!"

- 9** Výsledkem výrazu  $(0 \parallel A)$ , kde A je proměná typu integer je:  
**A** 1  
**B** 0 pokud je A rovno 0  
**C** 0  
**D** není definováno

- 10** Hodnota proměnné x po provedení následující kódu bude:  
int x = 0;  
if (1)  
    x = x + 1;  
else  
    x = x - 1;  
for (int i = 0; i < 100; i++) {x++};  
**A** 1  
**B** -1  
**C** 0  
**D** 101  
**E** kód se nepřeloží (obsahuje chybu)

- 
- 11** Po provedení kódu:  
`int A = 100;  
int* pA = &A;  
A = *A + 200;`  
Bude obsah proměnné A roven:
- A** 200
  - B** 300
  - C** kód obsahuje syntaktickou chybu
  - D** 100
  - E** obsah hodnoty A nelze bez spuštění jednoznačně určit
- 
- 12** Cyklus  
`int i = 0;  
while (i < 100) {  
 i++;  
}`  
proběhne:
- A** 99x
  - B** 100x
  - C** kód obsahuje syntaktickou chybu
  - D** 101x
- 
- 13** Následující kód reprezentuje:  
`char text[] = "Ahoj!";`
- A** Pole o 6 znacích a textový řetězec o délce 5 znaků
  - B** Pole o 5 znacích a textový řetězec o délce 5 znaků
  - C** Pole o 5 znacích a textový řetězec o délce 6 znaků
  - D** kód se nepřeloží (obsahuje chybu)
  - E** Pole o 6 znacích a textový řetězec o délce 6 znaků
- 
- 14** Datový typ `int` je:
- A** Typ bez znaménka pro uchovávání celých čísel
  - B** Typ bez znaménka pro uchovávání čísel s desetinnou čárkou
  - C** Typ se znaménkem pro uchovávání celých čísel
  - D** Typ se znaménkem pro uchovávání čísel s desetinnou čárkou
-