

PB071: pb071_vnitro_13hod

Jméno a příjmení – pište do okénka	UČO	Číslo zadání
		1

Více odpovědí může být správných. Za zcela správně zodpovězenou otázku máte 2 body, lze získat i část bodů za část správných odpovědí. Za každou chybnou odpověď je -1 bod.

1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int** foo() {
    const int size = 3;
    int** array = 0;
    array = malloc(sizeof(int*) * size);
    for (int i = 0; i < size; i++) array[i] = malloc(sizeof(int) * size);
    return array;
}
int main(void) {
    int** myArray;
    myArray = foo();
    free(myArray);
    return 0;
}
```

Výše uvedený program:

- A** Způsobí memory leak
- B** Dynamicky naalokuje pravoúhlé (ve všech rozměrech stejný počet prvků) trojrozměrné pole do ukazatele array
- C** Dynamicky naalokuje nepravoúhlé (v různých rozměrech může mít různý počet prvků) dvojrozměrné pole do ukazatele array
- D** Nelze přeložit
- E** Dynamicky naalokuje nepravoúhlé (v různých rozměrech může mít různý počet prvků) trojrozměrné pole do ukazatele array
- F** Dynamicky naalokuje pravoúhlé (ve všech rozměrech stejný počet prvků) dvojrozměrné pole do ukazatele myArray

2

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main() {
    char string[10];
    memset(string, 0, 10);
    strcpy(string, "Test");
    int size1 = sizeof(string);
    int size2 = strlen(string);
    int size3 = 0;
    for (int i = 0; i < 10; i++) {
        if (string[i] == 0) size3++;
    }
    return 0;
}
```

Pro obsah proměnných size1, size2 a size3 těsně před ukončením funkce main() platí:

- A** Hodnota proměnné size1 bude sizeof(char*)
- B** Hodnota proměnné size1 bude 10
- C** Hodnota proměnné size3 bude 6
- D** Hodnota proměnné size3 bude 10
- E** Hodnota proměnné size2 bude 5
- F** Hodnota proměnné size2 bude 10

```
3 #include <stdio.h>
  struct X { int value; };
  void foo(struct X a1, struct X a2) {
    a1.value += a2.value;
    a2.value = a1.value;
  }

  int main() {
    struct X a1 = {1};
    struct X a2 = {2};
    foo(a1, a2);
    printf("%d %d", a1.value, a2.value);
    return 0;
  }
```

Výstupem tohoto programu bude:

- A program nelze přeložit
- B žádná z ostatních odpovědí není správná
- C 1 2
- D 3 3
- E 3 2

```
4 #include <stdio.h>
  int main() {
    unsigned int flags = 0;
    flags = 0x03;
    flags = flags << 2;
    for (unsigned char i = 1; i <= 8; i = i * 2) {
      if (flags & i) printf("%d ", i);
    }
    return 0;
  }
```

Výše uvedený program:

- A Vypíše 1 2 3 4 5 6 7 8
- B Vypíše 1 2
- C Vypíše 4 8
- D Žádná z ostatních odpovědí není správná
- E Vypíše 2 4
- F Vypíše 3

```
5 int array[] = {1, 2, 3, 4};
  int main(void) {
    int value = 0;
    *(array + 0) = value;
    return 0;
  }
```

Překladač vypsal během překladač výše uvedeného programu následující:
..\main.c: In function 'main':
..\main.c:4:11: error: incompatible types when assigning to type 'int[4]' from type 'int'
Které z uvedených změn kódu umožní provedení překladač bez chyb?

- A přetypovat přiřazovanou hodnotu value na int[4]: *(array + 0) = (int[4])value;
- B neprovádět dereferenci proměnné array před přiřazením: array + 0 = value;
- C dereferencovat proměnnou value před přiřazením do array: *(array + 0) = *value;
- D specifikovat index položky: *(array + 0)[0] = value;
- E deklarovat pole array jako array{4} namísto array[]
- F k vypsání této chyby překladačem nemohlo dojít

```
6 #include <stdio.h>
  int main() {
    float a = (float) (9 / 10);
    printf("%f", a);
    return 0;
  }
```

Výše uvedený program vypíše:

- A nelze přeložit
- B 1.000000
- C 0.000000
- D žádná z ostatních odpovědí není správná
- E 0.990000

7

```
#include <stdio.h>
int array[] = {1, 2, 3, 4};
void foo() {
    int* pVal = array;
    pVal++;
    printf("%d ", *pVal);
}
int main(void) {
    foo();foo();foo();
    return 0;
}
```

Výše uvedený program vypíše:

- A** 1 1 1
 - B** 1 2 3
 - C** nelze předem určit, záleží na aktuálních hodnotách v poli array
 - D** nelze přeložit
 - E** žádná z ostatních odpovědí není správná
-

8

```
void foo(int a) {
    while (a < 10) {
        if (a > 8) break;
        if (a > 7) continue;
        a++;
        printf("%d ", a);
    }
}
```

Zavolání funkce foo(3):

- A** vypíše na standardní výstup 5 6 7 8
 - B** vypíše na standardní výstup 4 5 6 7 8 9
 - C** žádná z ostatních odpovědí není správná
 - D** vypíše na standardní výstup 5 6 7 8 9
 - E** Způsobí zacyklení programu
-

9

```
#include <stdlib.h>
int main() {
    char* array = malloc(10);
    char* array2 = array;
    for (char i = 0; i < 10; i++) array[i] = i;
    *(array - 1) = 1;
    return 0;
}
```

Pro uvedený kód platí:

- A** Program nelze přeložit
 - B** Program zapíše za konec alokovaného pole
 - C** Program zapíše před začátek alokovaného pole
 - D** Program neprovádí dealokaci veškeré alokované paměti
 - E** Program se zacyklí
 - F** Program vyplní všechny položky alokovaného pole
-

10

```
#include <stdlib.h>
#include <stdio.h>
void foo(int** array) {
    array = malloc(40);
    *array = NULL;
}
int main() {
    int* array = NULL;
    foo(&array);
    if (array != NULL) printf("Hello");
    else printf("World");
    free(array);
    return 0;
}
```

Výše uvedený program:

- A** způsobí memory leak o velikosti $40 * \text{sizeof}(\text{char})$
- B** vypíše World
- C** nezpůsobí žádný memory leak
- D** vypíše Hello
- E** žádná z ostatních odpovědí není správná