

ProgramowanieProceduralne

[Strona główna](#) / [Moje kursy](#) / [PP](#) / [LAB 4](#) / [IS_L9](#)

IS_L9

Przykład:

```
#include <stdio.h>

int * zmienna(void){
    int * zm;
    zm = malloc (sizeof (int) * 1);

    if (zm != NULL)
        *zm = 678;
    else
        printf ("nie udalo sie\n");
    return zm;
}

int main(void){
    int *var;
    var = zmienna();
    if (var != NULL)
        {printf ("wartosc = %d\n", *var);
        free (var);
        }
    return 0;
}
```

WE WSZYSTKICH ZADANIACH PROSZĘ PAMIĘTAĆ O SPRAWDZENIU POPRAWNOŚCI PRZYDZIELENIA PAMIĘCI ORAZ O ZWALNIANIU PAMIĘCI

1. (1) Proszę napisać program, który alokuje pamięć dla jednowymiarowej tablicy zmiennych rzeczywistych o rozmiarze podanym przez użytkownika, wypełnia ją liczbami pseudolosowymi z przedziału od 23.0 do 34.0. Tablicę proszę wypisać przy użyciu zmiennej ***iter**. Następnie program ma wczytać nowy rozmiar tablicy i gdy jest on większy niż dotychczasowy nowe elementy należy wypełnić liczbami z zakresu od -20.0 do -5.0. Na koniec ponownie proszę wypisać tablicę przy użyciu zmiennej ***iter**.

2. (2) Proszę napisać i przetestować funkcję, która przyjmuje jako parametry 2 wskaźniki na liczby rzeczywiste i oblicza podłogę (**floor**) z iloczynu wartości, na które wskazują te wskaźniki. Funkcja ma zwrócić wskaźnik na zmienną całkowitą przechowującą tak otrzymaną wartość.

3. Trójkąt Pascala dla N = 5

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

(4) Proszę napisać program, który wypisze na ekran trójkąt Pascala o podanej przez użytkownika wysokości.

W programie korzystamy **TYLKO z JEDNEJ** tablicy **jednowymiarowej**, która przechowuje **aktualnie** wypisywany poziom.

Na początku programu tablica przechowuje tylko jeden element: **pas[0]=1**.

W każdej kolejnej iteracji **zwiększamy rozmiar tablicy** o jeden element, wykorzystaj funkcję **realloc**. Zawsze **pas[0]=1** oraz **pas[ostatni element]=1**.

4. (3) Proszę napisać program, w którym dwie posortowane rosnąco tablice liczb rzeczywistych zostaną połączone w nową tablicę dynamiczną, która będzie posortowana rosnąco, wartości będą pochodziły z przekazanych tablic i nie będą się powtarzały.

UWAGA

Tablica **wynik** ma "rosnąć" wraz z dodawaniem do niej elementów. Nie należy tworzyć tablicy **wynik** o rozmiarze równym sumie rozmiarów tablic **arg_1** oraz **arg_2**, a dopiero później sortować i usuwać elementy powtarzające się.

Należy skorzystać z faktu, że **arg_1** oraz **arg_2** są posortowane

przykład :






```
float arg_1[] = {1.0, 1.2, 1.2, 1.5, 1.5, 1.5, 1.6, 1.6, 1.8, 1.9}
float arg_2[] = {1.0, 1.1, 1.1, 1.2, 1.4, 1.5, 1.6, 1.6, 1.8, 1.8, 1.8, 1.9, 1.9, 2.0, 2.0}
wynik -> {1.0, 1.1, 1.2, 1.4, 1.5, 1.6, 1.8, 1.9, 2.0}
```

5. (3) Proszę napisać program , w którym zostanie zaalokowana tablica liczb rzeczywistych o rozmiarze podanym przez użytkownika, zostanie wypełniona liczbami losowymi z zakresu (-1.5, 1.5), a następnie jej elementy zostaną "rozłożone " na dwie tablice dynamiczne : dodatnią oraz ujemną

UWAGA

Podobnie jak w zadaniu poprzednim tablice wynikowe mają zwiększać rozmiar wraz z dodawaniem do nich elementów.

Status przesłanego zadania

Status przesłanego zadania	Przesłane do oceny
Stan oceniania	Nieocenione
Termin oddania	poniedziałek, 16 marca 2020, 14:25
Pozostały czas	Zadanie zostało złożone 4 min. 46 sek. przed terminem
Ostatnio modyfikowane	poniedziałek, 16 marca 2020, 14:20
Przesyłane pliki	<div> <div>-  zad1.c</div> <div>16 marca 2020, 14:20</div> <div>-  zad2.c</div> <div>16 marca 2020, 14:20</div> <div>-  zad3.c</div> <div>16 marca 2020, 14:20</div> <div>-  zad4.c</div> <div>16 marca 2020, 14:20</div> <div>-  zad5.c</div> <div>16 marca 2020, 14:20</div> </div>
Komentarz do przesłanego zadania	<div> <div>▶ Komentarze (4)</div> </div>

◀ WYKŁAD 3

Przejdź do...

IS_L9 ▶





Platforma e-Learningowa obsługiwana jest przez:
Centrum e-Learningu AGH oraz Centrum Rozwiązań Informatycznych AGH

[Pobierz aplikację mobilną](#)