

# ProgramowanieProceduralne

[Strona główna](#) / [Moje kursy](#) / [PP](#) / [LAB 5](#) / [IS\\_L9](#)

## IS\_L9

Mamy procedurę wypisującą tablicę dwuwymiarową :

```
void print_tab(int a[][4], int x){
    int i, j;
    for (i = 0; i < x; i++){
        for (j = 0; j < 4; j++){
            printf ("%3d", a[i][j]);
            printf ("\n");
        }
    }
    return;
}
```

oraz procedurę wypisującą tablicę jednowymiarową

```
void wypisz_i (int *poczatek, int *koniec) {
    while (poczatek < koniec)
        printf ("%4d", *poczatek++);
    printf ("\n");
    return;
}
```

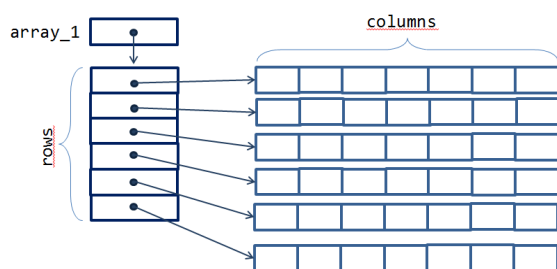
### 1. W zadaniu proszę korzystać z identyfikatorów `array_1`, `array_2`, `array_3` oraz nie korzystać z operatora `&`

Proszę zaalokować i wyzerować zawartość (`calloc`) trzech implementacje dynamicznej tablicy dwuwymiarowej `array_1`, `array_2`, `array_3` o elementach typu `int`

Każdą tablicę należy wypisać wierszami z wykorzystaniem procedury `wypisz_i`, a następnie zwolnić pamięć

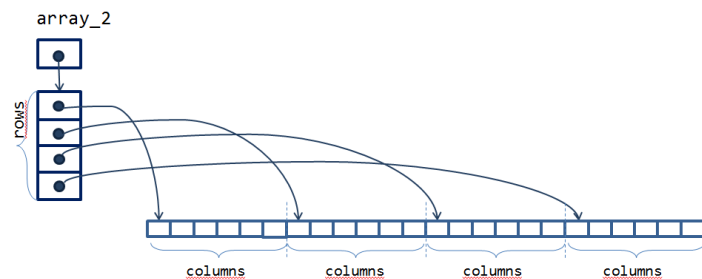
**rows i columns** zdefiniowane przez `#define`

- (1.5) dynamiczna tablica `array_1`, która ma **rows** elementów. Do każdego elementu tablicy `array_1` przypięty jest (dynamiczny) wiersz o długości **columns**. Wiersze nie tworzą spójnego obszaru pamięci. Organizacja pamięci wyglądała jak na rysunku:



Po wypisaniu zawartości **PROSZĘ ZWOLNIĆ PAMIĘĆ**

- (1.5) dynamiczna tablica `array_2`, która ma **rows** elementów. Do pierwszego elementu tablicy `array_2` przypięty jest (dynamiczny) spójny obszar pamięci dla wszystkich wierszy. Każdy wiersz ma długość **columns**. Kolejne elementy tablicy `array_2` wskazują na pierwszy element kolejnego wiersza. Organizacja pamięci wyglądała jak na rysunku:



Po wypisaniu zawartości PROSZĘ ZWOLNIĆ PAMIĘĆ

- (1) dynamiczna tablica `array_3` o rozmiarach **rows x columns**, "przypięta" do wskaźnika `int (* array_3)[columns]`

Po wypisaniu zawartości PROSZĘ ZWOLNIĆ PAMIĘĆ

2.

(1) Proszę uzupełnić program

```
#include <stdio.h>

#include <stdlib.h>
#include <string.h>

void print_string_array(_____, _____); //procedura wypisująca tablicę stringów
_____ min_string (_____, _____); //znajdowanie najmniejszego stringu w sensie leksykalnym w tablicy stringów - zwraca wskaźnik do
minimalnego

int main()
{
    _____ strings[] = { "Zorro", "Alex", "Celine", "Bill", "Forest", "Dexter"};

    _____ wsk_string;

    size_t strings_len = _____; //okreslenie długości tablicy strings

    print_string_array(strings, strings_len); //wypisanie tablicy stringów

    wsk_string = min_string(strings, strings_len); //znalezienie najmniejszego stringu w tablicy

    printf("_____\n", wsk_string); //wypisanie najmniejszego stringu

    return 0;
}
```

3. Przed przystąpieniem do pisania, proszę przeczytać treść zadania do końca!!!! - końcowe polecenia mają wpływ na sposób realizacji zadania

dobrym pomysłem jest wykorzystanie typedef

(4) Proszę napisać funkcję `mn_tab_1`, która mnoży dwie tablice A i B (mnożenie macierzy nie elementów tablicy).

W funkcji `main` deklarujemy dwie tablice `int t_A` oraz `int t_B`, które mnożymy za pomocą napisanej funkcji i wynik zapisujemy do zmiennej `t_C`

```
int t_A [2][3] = {1,2,3,4,5,6};
int t_B [3][4] = {1,1,1,1,1,1,1,1,1,1,1,1};
```

```
.....t_C = mn_tab_1 (A, B, 2);
```

Funkcja `mn_tab_1` ma być tak skonstruowana, by tablicę podpiętą do zmiennej `t_C` można było wypisać przy użyciu procedury `void print_tab(int (*)[4], int )`

Wynik proszę wypisać zarówno przy użyciu procedury `void print_tab(int (*)[4], int );` jak i `wypisz_i (int *, int *)`;

#### 4. W programie nie korzystamy ze zmiennych globalnych.

##### Zadanie nr1, może pomóc w realizacji funkcji.

Jeżeli do wskaźnika "podpięta" jest tablica dynamiczna to jej rozmiar - ilość elementów musimy znać (czyli musi być zmienna, która tę wartość przechowuje). Nie da się wyliczyć rozmiaru tablicy dynamicznej, tak jak w przypadku "zwykłej" tablicy

```
int *wsk = malloc(sizeof(int) * k);
int tab[k];
sizeof(wsk) - zwraca rozmiar wskaźnika do int (==sizeof(int*))
sizeof(tab) - zwraca k*sizeof(int)
```

(4) Proszę napisać funkcję zgodną z prototypem





```
int** create_table(int *)
```

która wczytuje liczby całkowite, aż do momentu gdy wprowadzimy 0.

Liczby mają utworzyć dynamiczną tablicę dwuwymiarową o wierszu długości 4. Tablica ma zostać wypisana wierszami w funkcji `main` z wykorzystaniem procedury `wypisz_i`

Brakujące wartości proszę wyzerować

## Status przesłanego zadania

<b>Status przesłanego zadania</b>	Przesłane do oceny
<b>Stan oceniania</b>	Nieocenione
<b>Termin oddania</b>	poniedziałek, 23 marca 2020, 14:25
<b>Pozostały czas</b>	Zadanie zostało złożone 6 min. 32 sek. przed terminem
<b>Ostatnio modyfikowane</b>	poniedziałek, 23 marca 2020, 14:18
<b>Przesyłane pliki</b>	<div> <div>-  <a href="#">zad1.c</a> 23 marca 2020, 14:18</div> <div>-  <a href="#">zad2.c</a> 23 marca 2020, 14:18</div> <div>-  <a href="#">zad3.c</a> 23 marca 2020, 14:18</div> <div>-  <a href="#">zad4.c</a> 23 marca 2020, 14:18</div> </div>
<b>Komentarz do przesłanego zadania</b>	<div> <div>► <a href="#">Komentarze (0)</a></div> </div>



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

[Pobierz aplikację mobilną](#)