

# ProgramowanieProceduralne

[Strona główna](#) / [Moje kursy](#) / [PP](#) / [LAB\\_12](#) / [LAB\\_9](#)

## LAB\_9

1.

( 3 ) Proszę utworzyć strukturę `struct Klient`, zawierającą pola: `f_name` i `l_name` do przechowywania imienia i nazwiska, pole `code` do przechowywania kodu pocztowego w rzeczywistym formacie oraz pole `age`.

Proszę napisać funkcję, która utworzy i zwróci tablicę jednowymiarową struktur `struct Klient`.

Argumentem funkcji ma być adres zmiennej przez którą prześlemy do `main` rozmiar powstałej tablicy oraz nazwa pliku tekstowego , który zawiera wartości odpowiadające polom struktury `struct Klient`. Zakładamy, że podany plik zawiera poprawną ilość wartości w odpowiednie j kolejności (imię, nazwisko, kod,wiek),natomiast ilość zapisanych struktur jest dowolna.

Następnie, proszę przetestować napisaną funkcję na pliku `spis`.

W kolejnym kroku, proszę zapisać tablicę do pliku binarnego `dane.dat`, a potem odczytywać po jednej strukturze (nie tworzymy tablicy) z pliku `dane` i wypisywać na ekran dane osób pełnoletnich.

2.

( 1 ) Proszę napisać i przetestować :

- makrodefinicję `MAK_1`, która będzie przyjmowała jako parametr wyrażenie typu `int`, a jej użycie spowoduje wypisanie na standardowym wyjściu tego wyrażenia (literalnie) oraz jego wartości.

*Przykład działania:*

```
int x=5, y=9;
MAK_1(7*x+2-y/6);
// w wyniku wywołania tego makra na ekranie powinno się pojawić
7*x+2-y/6=36
```

- jednoparametrową makrodefinicję `MAK_2`, której wartością jest
  - `1`, jeżeli argumentem jest liczba parzysta dodatnia,
  - `-1` jeżeli argumentem jest liczba parzysta ujemna, a
  - `0`, jeżeli argument jest nieparzysty.

3.

( 3 ) Dana jest struktura

```
struct tnode {
    char value;
    struct tnode * next;
};
```

- Proszę napisać funkcję, która dodaje element na **początek** listy jednokierunkowej.

```
_____ dodaj_na_poczatek (_____,char val){
    // alokacja pamięci na jeden element listy
    // uzupełnij obydwie pola utworzonego elementu
    //jeżeli lista jest pusta to dodaj element do pustej listy
    // jeżeli nie jest pusta dodaj element na początek listy
    // pamiętaj o zachowaniu ciągłości pomiędzy kolejnymi elementami
};
```

- Proszę napisać procedurę, która wypisuje listę na ekran.
- W funkcji `main()` proszę utworzyć wskaźnik do początku listy `head = NULL`. Proszę wykorzystać funkcję `dodaj_na_poczatek`, aby dodać do pustej listy elementy: `'a'`, `'c'`, `'v'`, `'f'`, `'t'`.
- Proszę wypisać listę na ekran
- Proszę napisać funkcję zwalnającą pamięć zajmowaną przez listę i wykorzystać ją do zwolnienia pamięci zajmowanej przez utworzoną listę

4.

(3) Dla struktury z poprzedniego zadania proszę napisać **procedurę** `dodaj_na_koniec`, która dodaje element na **koniec** listy jednokierunkowej

- W funkcji `main()` proszę utworzyć wskaźnik do początku listy `head = NULL`. Proszę wykorzystać procedurę `dodaj_na_koniec`, aby dodać do pustej listy elementy: 'a', 'c', 'v', 'f', 't'.
- Proszę wypisać listę na ekran - można skorzystać z procedury z poprzedniego zadania
- Na koniec funkcją z poprzedniego zadania proszę zwolnić pamięć

5. (3)

- Program zapisuje do pliku binarnego `dane_1.dat` dowolną (wczytaną) ilość liczb typu `float`.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

float rand_f(float min, float max){
    float r = (float) rand()/RAND_MAX;
    return r*(max - min) + min;}

int main(){
    float tab[30];
    int i, x, y;

    srand(time(0));
    FILE *f_1 = fopen ("dane_1.dat","wb");
    int var;
    printf ("rozmiar = ");
    scanf ("%d", &var);
    for (i=0; i<var; i++)
        tab[i]=rand_f(0.0, 40.0);


    //zapisywanie do pliku binarnego dane_1.dat zawartosci tablicy tab
    fwrite (tab,sizeof(float),var, f_1);
    fclose(f_1);
    return 0;
}
```

- Proszę dopisać do programu **funkcję** `.....matrix(int rows, int col, FILE fp)`, której argumentem jest wskaźnik do pliku binarnego oraz rozmiary macierzy.  
Funkcja ma zwracać adres dynamicznie utworzonej tablicy dwuwymiarowej liczb typu `float`, o rozmiarach takich jak argumenty wywołania (`rows, col`).  
Jeżeli w pliku jest mniej wartości, niż potrzeba na wypełnienie tablicy o rozmiarach przekazanych do funkcji, brakujące elementy zerujemy.  
Jeżeli jest więcej to wykorzystujemy tyle ile potrzeba.  
Jeżeli utworzenie macierzy się nie powiodło funkcja zwraca `NULL`.
- W programie należy przetestować działanie funkcji `matrix`, czyli wypisać zawartość utworzonych tablic w funkcji `main`, dla różnych wariantów wielkości pliku oraz rozmiarów tablicy (za dużo elementów w pliku, za mało, dobra ilość ale układ ilości kolumn i rzędów różny, np. jeżeli w pliku mamy 30 elementów to testujemy dla układu 3 rzędy i 10 kolumn, oraz 6 rzędów i 5 kolumn)

## Status przesłanego zadania

<b>Status przesłanego zadania</b>	Przesłane do oceny
<b>Stan oceniania</b>	Nieocenione
<b>Termin oddania</b>	poniedziałek, 18 maja 2020, 14:25
<b>Pozostały czas</b>	Zadanie zostało złożone 2 min. 5 sek. po terminie
<b>Ostatnio modyfikowane</b>	poniedziałek, 18 maja 2020, 14:27

## Przesyłane pliki

- 	<a href="#">zad1.c</a>	18 maja 2020, 14:27
- 	<a href="#">zad2.c</a>	18 maja 2020, 14:27
- 	<a href="#">zad3i4.c</a>	18 maja 2020, 14:27
- 	<a href="#">zad5.c</a>	18 maja 2020, 14:27

Komentarz do  
przesłanego  
zadania[▶ Komentarze \(2\)](#)[◀ IS\\_L9](#)[Obsługa Listy Jednokierunkowej ▶](#)

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

[Pobierz aplikację mobilną](#)