

ProgramowanieProceduralne

[Strona główna](#) / [Moje kursy](#) / [PP](#) / [LAB_6](#) / [IS_L7](#)

IS_L7

char* strcat(char* dest, const char* src);

dodaje zawartość napisu **src** do napisu **dest** (konkatenacja); zwraca **dest**.

Użytkownik musi zadbać o to, by **dest** i **src** kończyły się znakiem **'\0'** i żeby obszar pamięci zarezerwowany pod adresem wskazywanym przez **dest** był wystarczający do pomieszczenia obu napisów (i kończącego znaku **'\0'**).

Pierwszy znak **'\0'** w napisie **dest** jest nadpisywany przez pierwszy znak **src** i poczynając od tej pozycji przekopiowywane są znaki z **src** aż do kończącego go znaku **'\0'** włącznie.

```
#include <stdio.h>
#include <string.h>
int main(){
char tab[25] = "Ala ma ";
char tab2[] = "kota i psa";
printf("%s",strcat(tab,tab2));}
}
```

Jeżeli definiujemy funkcję, której argumentem ma być tablica wielowymiarowa i chcemy , aby wymiar - rozmiar elementu tablicy był określony przez zmienną to argument reprezentujący ten rozmiar w definicji funkcji musi pojawić się przed deklaracją tablicy.

Jako przykład zadanie nr 3 z poprzedniego laboratorium (mnożenie macierzy) w wersji ogólnej, czyli dla dowolnie dużych tablic - oczywiście wymóg równości ilości kolumn tablicy pierwszej i ilości wierszy w tablicy drugiej, wynikający z definicji mnożenia macierzy pozostaje)

```

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

void print_tab(int a[][4], int x){//bedzie wypisywała tablice dwuwymiarowa o wierszach dlygich na 4 int'y
    int i, j;
    for (i = 0; i < x; i++){
        for (j = 0; j < 4; j++)
            printf ("%3d", a[i][j]);
        printf ("\n");
    }
    return;
}

void print_tab_all(int size,int a[][size], int x){//bedzie wypisywała tablice dwuwymiarowa o dowolnie dlugich wierszach
//dlugosc wiersza size - musi byc w nagłówku przed ( nie koniecznie bezpośrednio) deklaracją tablicy, k torej wiersze sa
dlugie na size
//ilosc wierszy x moze byc w dowolnym miejscu
    int i, j;
    for (i = 0; i < x; i++){
        for (j = 0; j < size; j++)
            printf ("%3d", a[i][j]);
        printf ("\n");
    }
    return;
}

//trzeci sposób ale tutaj trzeba zadbac, zeby zaalokowany obszar był spójny i dobrze zrzutować wskazniki
int * mn_tab_all(int size_1, int size_2, int(*A)[size_1], int(*B)[size_2], int n) { //mnozmy macierz A n x size1 i macierz
B size1 x size2
    int(*C)[size_2]=calloc(n*size_2, sizeof(int));// tablica, ktora powstanie z mnozenia bedzie miala wiersze dlugosci
size_2 razy int, a bedzie ich tyle ile wierszy w tablicy A

    for(int i=0; i<n; i++){
        for(int j=0; j<size_2; j++){
            for(int k=0; k<size_1; k++){
                C[i][j]+=A[i][k]*B[k][j];
            }
        }
    }
    return *C ; //ale zwracamy wskaznik do pojedynczego int
}

int main(){
    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};

    int (*t_E)[4] = (int (*)[4]) mn_tab_all (3,4,t_A, t_B, 2);

    printf("\n1.-----\n");
    print_tab(t_E, 2);
    printf("\na2.-----\n");
    print_tab_all(4,t_E, 2);
    free(t_E);
}

```

```

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

```

1. (3)

Proszę napisać i przetestować funkcję **concat**, do której przekazujemy tablicę stringów. Funkcja zwraca wskaznik - string, który jest sklejeniem kolejnych stringów z przekazanej tablicy w kolejności występowania . Wynik ma być wypisany w funkcji **main**

Dane do testowania : `char *strings[] = {"The sun did not shine.", "It was too wet to play.", " So we sat in the house - ", "All that cold, cold, wet day."}`

Efekt działania programu:

```
The sun did not shine.It was too wet to play. So we sat in the house - All that cold, cold, wet day.
```

2.(2)

W programie mamy trzy procedury - jedna gotowa, dwie trzeba napisać zgodnie z opisem

```
void swap(int*a, int*b){
int temp;
temp = *a;
*a = *b;
*b = temp;
return;
};
```

```
void swap_tab(int*a, int*b, int){
//procedura wymieniająca zawartość tablic a i b (o tym samym rozmiarze) przy użyciu procedury swap
// wymieniamy zawartość elementu a[i] z zawartością elementu b[i]
return;
};
```

```
void swap_row(int n, int (*a)[n], int x, int y){
//procedura wymieniająca wiersz x z wierszem y - wiersze mają tę samą długość
// przy użyciu procedury swap_tab
return;
};
```

Proszę przetestować napisane procedury na tablicach, tablice należy wypisać wierszami **przed** i **po** zamianie wierszy procedurą **wypisz_i**

- (0.5) `int tab_1[4][5] = {1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,4};` //- wymiana wiersza 1 z 3 przy użyciu procedury **swap_raw**
- (0.5) `int tab_11[4][5] = {1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4};` //- wymiana wiersza 0 z 3 przy użyciu procedury **swap_tab**
- (0.5) `int tab_2 [5][2] = {1,1,2,2,3,3,4,4,5,5};` //- wymiana wiersza 0 z 2 przy użyciu procedury **swap_raw**
- (0.5) `int tab_22 [5][2] = {1,1,2,2,3,3,4,4,5,5};` //- wymiana wiersza 1 z 2 przy użyciu procedury **swap_tab**
- (0.5) dla tablicy **tab_3** - wymiana wiersza 1 z 5

```
int **tab_3 = malloc( 6*sizeof(int*));

for(i = 0; i < 6; i++)
tab_3[i] = malloc(3 * sizeof(int));

for(i = 0; i < 6; i++)
    for(j = 0; j < 3; j++)
        tab_3[i][j] = rand() % 21;
```

- (0.5) dla tablicy **tab_33** - wymiana wiersza 2 z 4

```
int **tab_33= malloc( 6*sizeof(int*));

tab_33[0] = malloc(6*3 * sizeof(int));
for(i = 1; i < 6; i++)
tab_33[i] = tab_33[i-1]+3;

for(i = 0; i < 6; i++)
    for(j = 0; j < 3; j++)
        tab_33[i][j] = rand() % 21;
```

3.(4)

Proszę napisać procedurę **proc_1**, która przesuwa **cyklicznie** elementy tablicy jednowymiarowej typu **int** w lewo o zadaną ilość miejsc (mniejszą od ilości eementów tablicy)

Prcedurę proszę przetestować na tablicach:

- `int tab_1[5] = {1,2,3,4,5};` - wypisać **po** przesunięciu procedurą **wypisz_i**
- `int tab_2 [4][2] = {{10,11},{12,13},{14,15},{16,17}};` - wypisać wierszami **po** przesunięciu procedurą **wypisz_i**
- dwuwymiarowa tablica **dynamiczna** o 5 wierszach i 3 kolumnach wypełniona wartościami od 0 do 14 - wypisać wierszami **przed** i po przesunięciu procedurą **wypisz_i**

Status przesłanego zadania

Status przesłanego zadania	Przesłane do oceny
----------------------------	--------------------

Stan oceniania	Nieocenione		
Termin oddania	poniedziałek, 30 marca 2020, 14:25		
Pozostały czas	Zadanie zostało złożone 4 min. 19 sek. przed terminem		
Ostatnio modyfikowane	poniedziałek, 30 marca 2020, 14:20		
Przesyłane pliki	<div><div><div>-</div><div></div><div>1.c</div><div>30 marca 2020, 14:20</div></div><div><div>-</div><div></div><div>2.c</div><div>30 marca 2020, 14:20</div></div><div><div>-</div><div></div><div>3.c</div><div>30 marca 2020, 14:20</div></div></div>		
Komentarz do przesłanego zadania	<div><div>▶</div><div>Komentarze (1)</div></div>		

◀ LAB_6

Przejdź do...

funkcja strtok - warto się zaprzyjaźnić - przyda się ▶



Platforma e-Learningowa obsługiwana jest przez:
Centrum e-Learningu AGH oraz Uczelniane Centrum Informatyki AGH

Podsumowanie zasad przechowywania danych
[Pobierz aplikację mobilną](#)