

TABLICE

1. Tablice jednowymiarowe

Deklaracja

```
typ nazwa_tablicy[rozmiar];  
  
int tablica[20];
```

Inicjalizacja, czyli nadawanie wartości

```
int tablica[3] = {1,2,3};  
int tablica[20] = {1};  
int tablica[] = {1, 2, 3, 4, 5};  
int Tab[100]={0};
```

```
#include <stdio.h>  
#define ROZMIAR 3  
  
int main()  
{  
    int tab[ROZMIAR] = {3,6,8};  
    int i;  
    printf("Druk tablicy tab:\n");  
    for (i=0; i<ROZMIAR; ++i)  
    {  
        printf("Element numer %d =  
        %d\n", i, tab[i]);  
    }  
    return 0;  
}
```

Wynik:

Druk tablicy tab:
Element numer 0 = 3
Element numer 1 = 6
Element numer 2 = 8

```
//to samo, trochę inaczej...  
#include <stdio.h>  
int main()  
{  
    int tab[3] = {3,6,8};  
    int i;  
    printf("Druk tablicy tab:\n");  
    for (i=0; i<(sizeof tab / sizeof *tab); ++i)  
    {  
        printf("Element numer %d = %d\n", i, tab[i]);  
    }  
    return 0;  
}
```

2. Odczyt/zapis wartości do tablicy

Z tablicami posługujemy się tak samo jak ze zwykłymi zmiennymi. Różnica polega jedynie na podaniu **indeksu** tablicy. Indeksy numerowane są od **zera!!!**

```
.....  
int tablica[5] = {0};  
int i = 0;  
  
tablica[2] = 3;  
tablica[3] = 7;  
  
for (i=0;i!=5;++i)  
    printf("tablica[%d]=%d\n", i, tablica[i]);  
.....
```

3. Tablice znaków czyli łańcuchy

Tablice znaków tj. typu char oraz unsigned char posiadają dwie ogólnie przyjęte nazwy, zależnie od ich przeznaczenia:

- bufor — gdy wykorzystujemy je do przechowywania ogólnie pojętych danych, gdy traktujemy je jako po prostu "ciągi bajtów" (typ char ma rozmiar 1 bajta, więc jest elastyczny do przechowywania np. danych wczytanych z pliku przed ich przetworzeniem).
- napisy — gdy zawarte w nich dane traktujemy jako ciągi liter lub „łańcuchy”.

```
char *tekst = "Jakiś tam tekst";  
printf("%c\n", "przykład"[0]); /* wypisze p - znaki w napisach są numerowane od zera */  
printf("%c\n", tekst[2]); /* wypisze k */
```

Jak wczytać łańcucha

`gets(nazwa łańcucha)`

`scanf(„%s”, nazwa łańcucha)` – tylko jeżeli jest to tablica o zdefiniowanej długości!!!

Operacje na tekstach:

`strcat(s,t)` dopisuje **n** znaków z **t** na koniec **s**

`strncat(s,t,n)` dopisuje **t** na koniec **s**

`strcmp(s,t)` zwraca wartość ujemną zero lub wartość dodatnią odpowiednio dla: $s < t$, $s = t$, $s > t$

`strncmp(s,t,n)` to samo, tylko dla pierwszych **n** znaków

`strcpy(s,t)` kopiuje **t** do **s**

`strncpy(s,t,n)` kopiuje **n** znaków z **t** do **s**

`strlen(s)` zwraca długość **s**

`strchr(s,c)` zwraca wskaźnik do pierwszego wystąpienia **c** w **s** lub NULL gdy nic

`strrchr(s,c)` zwraca wskaźnik do ostatniego wystąpienia **c** w **s** lub NULL gdy nic

4. Tablice wielowymiarowe

Deklaracja: `typ nazwa_tablicy[wiersze] [kolumny];`

`float macierz[10][10];`

Inicjalizacja, czyli nadawanie wartości

`macierz[2][3] = 1.2;`

```
float macierz[3][4] = {
{ 1.6, 4.5, 2.4, 5.6 }, /* pierwszy wiersz */
{ 5.7, 4.3, 3.6, 4.3 }, /* drugi wiersz */
{ 8.8, 7.5, 4.3, 8.6 } /* trzeci wiersz */
};
```

UWAGA: `int tab[100];`
`int i;`
`for (i=0; i<=100; ++i) // powinno byc i<100`
`tab[i] = 0;`

Materiały te są skompletowane w oparciu o **Programowanie w C**, stworzone na Wikibooks, bibliotece wolnych podręczników.

Wydanie I z dnia 2 września 2008

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".