

# Tablice jednowymiarowe w C++

## Inicjacja tablicy jednowymiarowej

Tablice jednowymiarowe można porównać do osi liczbowej. Aby odwołać się do jakiegoś elementu, wystarczy podać jedną współrzędną.

Tablice tego typu deklarujemy następująco:

**typ\_komórek\_tablicy nazwa\_tablicy [ ilość\_elementów\_tablicy ] ;**

np.:

```
int tablica[1000];
```

czyli tablica, która może przechowywać tysiąc elementów typu całkowitego,

```
char znaki[1001];
```

czyli tablica, która może przechowywać 1000 znaków + znak końca tablicy.

## Nadanie wartości początkowych tablicy

Operację tą można zrobić tylko przy deklaracji tablicy (przy tworzeniu tablicy). Operację nadania wartości tablicy przedstawię na przykładzie pięcio-elementowej tablicy liczb całkowitych:

```
int tab[5] = {1, 3, 90, 100000, 9};
```

Oznacza to, że w pierwszej komórce przechowywana jest liczba **1**, w drugiej **3** itd..

Jeśli przypiszemy do tablicy mniejszą ilość elementów niż wielkość tablicy, to pozostałe komórki będą miały wartość zero:

```
int tab[5] = {1, 3};
```

Powyższy zapis jest równoważny z zapisem:

```
int tab[5] = {1, 3, 0, 0, 0};
```

Przypisanie do tablicy pustego nawiasu oznaczać będzie, że wszystkie komórki będą miały wartość zero.

Warto wiedzieć, że jeśli tablicę zadeklarujemy jako globalną, to elementy tablicy zostaną także wyzerowane.

## Odwoływanie się do komórek tablicy

Najważniejszą regułą jest to, że komórki **numerujemy od zera**. Numery te nazywamy **indeksami tablicy**. Oznacza to, że jeśli stworzyliśmy tablicę 10-elementową, to numer pierwszej komórki jest równy zero, drugiej jeden, ..., no i ostatniej dziewięć:

$$\underbrace{[0][1][2] \cdot \cdot \cdot [n-1]}_{\text{indeksy komórek n-elementowej tablicy}}$$

Odwołujemy się do komórek tablicy podając **jej indeks** w nawiasie kwadratowym, np:

```
int tab[10]; //deklaracja tablicy

tab[0] = 34; //przypisanie do pierwszej komórki wartość 34

tab[9] = 100; //przypisanie do ostatniej komórki wartość 100

cout<<tab[0]; //wyświetlenie zawartości pierwszej komórki
//tablicy (czyli 34)
```

Pamiętaj!!! Gdy odwołujesz się do ostatniej komórki tablicy n-elementowej, to jej indeks jest równy n-1.

### Przykładowe zadanie

**Zadanie.** Napisz program, który nada następujące wartości początkowe tablicy 5-elementowej: 1, 2, 5, 0, 0, a następnie wyświetli najpierw wartości parzyste tej tablicy, a następnie nieparzyste.

#### Rozwiązanie:

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main()
{
    int tab[5] = {1, 2, 5};

    for(int i=0;i<5;i++)
        //jeśli wartość tablicy jest parzysta
        if(tab[i]%2==0)
            cout<<tab[i]<<" ";

    for(int i=0;i<5;i++)
        //jeśli wartość tablicy jest nieparzysta
        if(tab[i]%2)
            cout<<tab[i]<<" ";

    cout<<endl;

    system("pause");
    return 0;
}
```

Warto zauważyć, że przeszukujemy tablicę o indeksach od 0 do 4, a więc w nawiasie kwadratowym tablicy wstawiliśmy licznik "i", który przyjmuje właśnie takie wartości.

# Tablice wielowymiarowe w C++

## Inicjacja tablicy dwuwymiarowej

Jako tablicę dwuwymiarową możemy sobie wyobrazić planszę prostokątną składającą się z pewnej liczby wierszy i kolumn (numerowanie zaczynamy od zera). Aby przypisać (pobrać) wartość do danej komórki, należy podać jej obie współrzędne.

Inicjacja tablicy polega na podaniu ilości wierszy i kolumn:

```
typ_elementów_tablicy nazwa_tablicy [ ilość wierszy][ilość kolumn ];
```

np.:

```
//stworzenie dwuwymiarowej tablicy liczb całkowitych, w sumie 100 komórek: 10x10.
```

```
int tab [ 10 ][ 10 ];
```

## Nadanie wartości początkowych tablicy

Wartości początkowe tablicy możemy nadać przy jej deklaracji. Popatrzmy na przykład oparty na tablicy dwuwymiarowej liczb całkowitych:

```
//zauważmy, że mamy trzy wiersze, i dwie kolumny
```

```
int tab[3][2] = { {1,3},{4,5},{0,-1} };
```

Przy nadawaniu wartości tablicy można pominąć wartość w pierwszym nawiasie kwadratowym:

```
//wartość pierwszego nawiasu została pominięta
```

```
int tab[][2] = { {1,3},{4,5},{0,-1} };
```

## Odwoływanie się do komórek w tablicy dwuwymiarowej

Aby odwołać się do każdej z komórek należy w nawiasach kwadratowych podać numer wiersza i kolumny komórki, do której się odwołujemy, pamiętając o tym, że **numerujemy je od zera**:

```
int tab[][2] = {{1,3},{4,5},{0,-1}}; //deklaracja i inicjacja tablicy dwuwymiarowej
```

```
cout<<tab[2][1]; //wyświetlenie wartości komórki znajdującej się w trzecim wierszu i w drugiej kolumnie (-1)
```

```
tab[0][0] = -100; //przypisanie do pierwszej komórki wartości -100
```

**Zad.** Napisz program, który wykona transpozycję macierzy 4x5. Liczby generujemy losowo z przedziału [-9; 9]. Elementami macierzy są liczby całkowite.

### Rozwiązanie:

Macierz to obiekt, który doskonale nadaje się do przechowywania w tablicach dwuwymiarowych. Każda macierz składa się z pewnej ilości wierszy i kolumn. Przykładowa macierz spełniająca warunki zadania:

Transpozycja macierzy polega na zamianie wierszy z kolumnami. Powyższa macierz powinna wyglądać następująco:

$$\begin{bmatrix} 1 & 2 & 3 & 2 \\ 1 & 2 & 3 & 4 \\ -5 & 0 & 6 & 5 \\ 8 & 0 & -6 & -9 \\ 2 & 5 & 4 & -8 \end{bmatrix}$$

```
#include<iostream>
using namespace std;

int main()
{
    int tab[4][5]

    cout<<"Przed transpozycja:\n ";
```

```

for(int i=0;i<4;i++)
{
    for(int j=0;j<5;j++)
    {
        //wygenerowanie liczb z zakresu [-9; 9]
        tab[i][j]=rand()%19-9;
        //wyświetlenie wylosowanej liczby
        cout<<tab[i][j]<<" ";
    }
    cout<<endl;
}

//transpozycja macierzy
cout<<"Po transpozycji: "<<endl;
for(int i=0;i<5;i++)
{
    for(int j=0;j<4;j++)
        cout<<tab[j][i]<<" ";

    cout<<endl;
}

system("pause");
return 0;
}

```

## Tablice wielowymiarowe

Tablice o większej liczbie wymiarów rzadko się stosuje. Sposób inicjacji, oraz operowania na tablicach tego typu jest analogiczny jak w przypadku tablic dwuwymiarowych.

Prześledźmy przykład tworzenia tablicy trójwymiarowej:

```

#include
#include<cstdlib>
using namespace std;

int main()
{
    //deklaracja tablicy trójwymiarowej
    //taka tablica posiada 3*4*5 = 60 komórek
    int tab[3][4][5];
    //przypisanie wartości 23 do pierwszej komórki
    tab[0][0][0] = 23;

    cout<<tab[0][0][0]<<endl;

    system("pause");
    return 0;
}

```

Odwołując się do komórek tablicy trójwymiarowej, możemy sobie wyobrazić, że odwołujemy się to jednostkowych sześciątów, z których zbudowany jest prostopadłościan. Aby "dostać się" do danego sześcianu, musimy określić jego współrzędne: długość, szerokość oraz wysokość.