

## Temat: Tablice w języku C++

# Porównanie języka C i C++:

---

W porównaniu obu języków skupimy się wyłącznie na różnicach pomiędzy komendami.

W języku C++ występuje tzw. przestrzeń nazw, stąd wiele z komend będzie się rozpoczynało od **std::**. Możemy pozbyć się tego przedrostka używając poniższej instrukcji po załączaniu bibliotek (linijka 3 Przykład 1):

`using namespace std;`

- ▶ Rozszerzenie plików
    - ▶ Pliki w języku C mają rozszerzenie .c, np.: **nazwa.c**
    - ▶ Pliki w języku C++ mają rozszerzenie .cpp, np.: **nazwa.cpp**
  - ▶ Standardowa biblioteka umożliwiająca operacji wejścia/wyjścia
    - ▶ Język C: **#include <stdio.h>**
    - ▶ Język C++: **#include <iostream>**
  - ▶ Wyświetlanie:
    - ▶ Język C: **printf("tekst");**
    - ▶ Język C++: **std::cout<<"tekst";**
  - ▶ Wczytywanie:
    - ▶ Język C: **scanf("%d", &i);**
    - ▶ Język C++: **std::cin>>i;**
- ▶ Podczas **Wyświetlania** i **Wczytywania** nie używamy kodów formatujących. Kompilator sam wie, jakiego typu jest zmienna.
  - ▶ Jeśli chcemy wyświetlić tekst łączony z wartością zmiennej korzystamy z operatora << np.:  
**int i=1;**  
**cout<<"Wartosc i = „" << i << endl;**
  - ▶ Instrukcja **endl**; oznacza przejście do następnej linii. Jest to odpowiednik \n.

Konstrukcja komentarzy, instrukcji warunkowych, pętli i funkcji pozostaje bez zmian.

Poniżej znajdują się dwa programy, jeden napisany w języku C i drugi w języku C++.

### Przykład 1:

Oba programy dadzą ten sam wynik:

```
1  #include <stdio.h>
2
3
4
5  int main () {
6      int a;
7      printf("Podaj a=");
8      scanf("%d", &a);
9      printf("%d\n", a);
10
11     double b;
12     printf("Podaj b=");
13     scanf("%lf", &b);
14     printf("%lf\n", b);
15
16     char c[50];
17     printf("Podaj c=");
18     scanf("%s", &c);
19     printf("%s\n", c);
20     return 0;
21 }
22
23
```

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main () {
6      int a;
7      cout<<"Podaj a=";
8      cin>>a;
9      cout<<a<<endl;
10
11     double b;
12     cout<<"Podaj b=";
13     cin>>b;
14     cout<<b<<endl;
15
16     string c;
17     cout<<"Podaj c=";
18     cin>>c;
19     cout<<c<<endl;
20     return 0;
21 }
22
23
```

## Tablice w języku C++:

---

- ▶ Tablica jest złożoną strukturą danych, składającą się z elementów tego samego typu.
- ▶ Dostęp do kolejnych elementów tablicy jest możliwy za pośrednictwem indeksów, czyli liczby określającej położenie elementu w tablicy.
- ▶ Numeracja elementów tablicy rozpoczyna się od 0.

## Tablice jednowymiarowe:

---

Definicja tablicy jednowymiarowej ma następującą postać:

- ▶ `typ nazwa_tablicy [rozmiar];`

## ► Przykład 2:

Przykład definicji tablicy jednowymiarowej:

```

1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      int a[5];
6      a[0] = 5;
7      a[3] = 7;
8      a[4] = a[3] - a[0];
9      cout<<a[4];
10     return 0;
11 }
12

```

a[0]	a[1]	a[2]	a[3]	a[4]
5			7	2

Komórkom tablicy możemy nadawać wartości pojedynczo tak jak w przykładzie powyżej lub podczas definiowania tablicy wpisując

► `int nazwa [6] = { 7, 10, -1, 0, 2, 5};`

## Przykład 3:

Wczytywanie i wypisywanie elementów tablicy.

stała N = 4

```

1  #include <iostream>
2  #define N 4
3  using namespace std;
4
5  int main()
6  {
7      int i;
8      double tab[N];
9      //wczytywanie elementow tablicy
10     for(i=0;i<N;i++) {
11         cout<<"Podaj tab["<<i<<"]=";
12         cin>>tab[i];
13     }
14     //wypisywanie elementow tablicy
15     cout<<"Wczytano elementy tablicy: ";
16     for(i=0;i<N;i++) cout<<tab[i]<<" ";
17     return 0;
18 }

```

```

Podaj tab[0]=1
Podaj tab[1]=3
Podaj tab[2]=9
Podaj tab[3]=7
Wczytano elementy tablicy: 1, 3, 9, 7,

```

#### Przykład 4:

Wykonywanie obliczeń na elementach tablicy.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int i, suma=0, suma_p=0, mmin, mmax;
7      int a[10]={1,2,3,4,5,6,7,8,9,10};
8      //suma elementow tablicy
9      for(i=0;i<10;i++) suma+=a[i];
10     //suma elementow o indeksach parzystych
11     for(i=0;i<10;i++) if(i%2==0) suma_p+=a[i];
12     //element minimalny
13     mmin=a[0];
14     for(i=1;i<10;i++) if(a[i]<mmin) mmin=a[i];
15     //element maksymalny
16     mmax=a[0];
17     for(i=1;i<10;i++) if(a[i]>mmax) mmax=a[i];
18     //wypisywanie podsumowania
19     cout<<"suma="<<suma<<" , suma_p="<<suma_p<<" , srednia=";
20     cout<<suma/10.0<<" , min="<<mmin<<" , max="<<mmax;|
21     return 0;
22 }
```

suma=55, suma p=25, srednia=5.5, min=1, max=10

#### Przykład 5:

Wypełnianie tablicy liczbami pseudolosowymi:

```
1  #include <iostream>
2  #include <ctime> //time()
3  #include <cstdlib> //rand()
4  using namespace std;
5
6  int main()
7  {
8      int i, losuj[100], ile=50;
9      srand((unsigned int)time(NULL)); //ustawienie zarodka
10     for(i=0;i<ile;i++) losuj[i]=rand()%100;
11     for(i=0;i<ile;i++) cout<<losuj[i]<<" , ";
12     return 0;
13 }
```

Funkcja `srand()` ustawia punkt startowy zależny od czasu dla funkcji `rand()`

`rand()` losuje liczbę od 0 do `RAND_MAX`, dlatego musimy użyć reszty z dzielenia, aby wylosowana liczba była w przedziale 0-99

# Tablice wielowymiarowe:

Definicja tablicy wielowymiarowej (np.: czterowymiarowej) ma następującą postać:

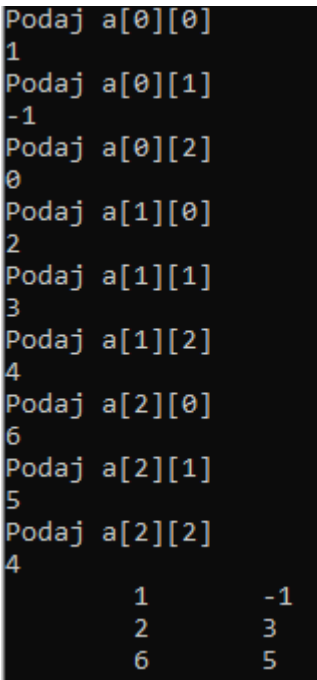
```
typ nazwa_tablicy [rozmiar1][rozmiar2][rozmiar3][rozmiar4];
```

Tablice wielowymiarowe rozumiemy jako uogólnienie wektora tj. tablica jednowymiarowa może być utożsamiana z wektorem, tablica dwuwymiarowa z macierzą itd.

## Przykład 6:

Zdefiniuj funkcję wyświetlającą macierz 3x3. Pobierz od użytkownika wartości komórek macierzy 3x3, następnie wypisz macierz.

```
1  #include <iostream>
2  #define N 3
3  using namespace std;
4
5
6  void wypisz_macierz(int m[N][N]) {
7      for (int i=0; i<N; i++) {
8          for(int j=0; j<N; j++) {
9              cout<<"\t"<< m[i][j] ;
10             }
11             cout<<endl;
12         }
13     }
14
15     int main () {
16         int a[N][N];
17         for (int i=0; i<N; i++) {
18             for(int j=0; j<N; j++) {
19                 cout<<"Podaj a["<<i<<"] ["<<j<<"] "<<endl;
20                 cin>>a[i][j];
21             }
22         }
23         wypisz_macierz(a);
24         return 0;
25     }
26
```



```
Podaj a[0][0]
1
Podaj a[0][1]
-1
Podaj a[0][2]
0
Podaj a[1][0]
2
Podaj a[1][1]
3
Podaj a[1][2]
4
Podaj a[2][0]
6
Podaj a[2][1]
5
Podaj a[2][2]
4
      1      -1      0
      2       3       4
      6       5       4
```

## Zadania do samodzielnej realizacji:

---

1. (1p) Napisz funkcję, która sprawdzi, czy wprowadzona przez użytkownika macierz wymiaru  $2 \times 2$  jest odwracalna, jeśli tak, to program obliczy i wyświetli macierz odwrotną.
2. (2p) Wczytaj liczbę  $m$  zapisaną w systemie ósemkowym i wypisz na ekranie tę liczbę w systemie dziesiętnym.  
*Wskazówka: Zamień liczbę 123 na tablicę  $[1, 2, 3]$ , a następnie oblicz  $1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0$ .*
3. (2p) Zadeklaruj dwie macierze o rozmiarze  $3 \times 3$ . Wypełnij je całkowitymi liczbami pseudolosowymi z zakresu od 0 do 1000. Oblicz sumę obu macierzy. Na ekranie wyświetl obie macierze i obliczoną sumę tych macierzy.