

Tablice dwuwymiarowe

Tablice dwuwymiarowe w C++ w zasadzie działają podobnie do tablic jednowymiarowych. Różnica między tablicami jednowymiarowymi, a dwuwymiarowymi jest w indeksach – tablice jednowymiarowe działają jak lista, gdzie był jeden indeks, który określał położenie wartości na liście:

n=0	n=1	n=2	n=3	n=4	n=5	n=6
4	12	6	33	99	7	41

Wartości **n** są indeksami, jak można zauważyć indeksy zaczynają się zawsze od 0. Więc gdy w tym wypadku mamy tablicę o rozmiarze **7** z indeksami od **0** do **6**. Gdyby nasza tablica miała rozmiar na przykład **40** to jej indeksy były by od **0** do **39**. Możemy bardzo prosto trawersować po tablicy przy użyciu pętli **for** np.:

```
for(int i = 0; i < r; i++){  
    if(i%2==0){  
        cout<<tab[i]<<endl;  
    }  
}
```

Działanie pętli wygląda następująco: zmienna **i** jest inkrementowana z każdym przejściem, pętla będzie się wykonywać, dopóki **i** jest mniejsze od **r**, gdzie **r** oznacza rozmiar tablicy.

Test logiczny **IF** będzie wypisywał co drugą wartość tablicy (dzielimy indeks przez 2 i patrzymy czy reszta z dzielenia = 0).

Dla przypomnienia, tablicę jednowymiarową deklarujemy w następujący sposób:

```
typ_zmiennej nazwa_tablicy[rozmiar];
```

Tak samo jak deklaracja zmiennej, tablica może mieć podane wartości od razu przy deklaracji lub później w dalszym kodzie. Możemy też określić rozmiar tablicy podanymi wartościami np.:

```
int calkowite[] = {1,2,3,4,5}
```

Dzięki temu zadeklarujemy tablicę typu **int** o nazwie **calkowite**, w której będą przechowane wartości 1, 2, 3, 4, 5.

Tablica dwuwymiarowa działa analogicznie do tablicy jednowymiarowej z wyjątkiem tylko że mamy teraz dwie osie (dwa wymiary) indeksów.

	X=0	X=1	X=2
Y=0	77	23	51
Y=1	18	16	79
Y=2	908	111	4

Jak widać, tablica dwuwymiarowa wygląda całkiem podobnie do układu współrzędnych, z tym że oś Y kieruje się w dół zamiast w górę. Gdybyśmy chcieli wypisać wszystkie elementy tej tablicy za pomocą pętli, musielibyśmy użyć dwóch pętli – jedna dla X, druga dla Y; albo jedna dla kolumn, a druga dla wierszy:

```
for(int y = 0; y < r; y++){           //Wiersze tablicy
    for(int x = 0; x < r; x++){        //Kolumny tablicy
        cout<<tab[y][x];
    }
    cout<<endl;
}
```

Jak widać pierwsza pętla (ze zmienną y) odpowiada za wiersz tablicy, a druga – wewnątrz pierwszej, odpowiada za kolumny.

Y0X0	Y0X1	Y0X2
Y1X0	Y1X1	Y1X2
Y2X0	Y2X1	Y2X2

Tak wyglądają poszczególne indeksy lub koordynaty dla poszczególnych komórek w tablicy.

Zadania

W każdym zadaniu, na początku kodu programu, dodaj swoje imię i nazwisko

1. Przy użyciu **pętli** i **tablic** przechowujących liczby całkowite wprowadź poniższe dane:

```

tab[0,0] = 0
tab[0,1] = 1
tab[0,2] = 2
tab[1,0] = 3
tab[1,1] = 4
tab[1,2] = 5

```

Przyda ci się zmienna licznikowa, która będzie iterowana z każdym przejściem pętli.

2. Przy użyciu **cout**, zwizualizuj tablicę z punktu 1. Wyjście w konsoli powinno wyglądać tak:

```
0 1 2
3 4 5
```

3. Przy pomocy funkcji **rand() % 100** z biblioteki **<cstdlib>** utwórz tablicę dwuwymiarową, której rozmiar będzie definiowany zmienną o wartości podanej przez użytkownika. Funkcją **rand** wypełnij komórki tablicy pseudolosowymi liczbami z zakresu od 0 do 99.

Przykładowy output konsoli:

```
Podaj rozmiar tablicy:
5
```

```
Wygenerowana tablica:
54 77 12 43 10
23 76 12 45 1
79 99 12 44 2
18 12 11 9 0
51 23 44 1 28
```

Możesz użyć „\t” w **cout** do wstawiania tabulacji, które poprawią wygląd tablicy.

4. Dodaj do kodu z punktu 3. sumowanie wierszy, które dodaje wszystkie wartości z jednego wiersza i wyświetla ich sumę. Na przykład:

```
54 77 12 43 10    = 196
23 76 12 45 1      = 157
79 99 12 44 2      = 236
18 12 11 9 0       = 50
51 23 44 1 28      = 147
```

Przyda ci się zmienna pomocnicza, która będzie przechowywać wynik dodawania wartości z wiersza.

Po zakończeniu, każdemu plikowi z kodem źródłowym (.cpp) z każdego zadania, zmień nazwę na numer zadania, czyli: 1.cpp, 2.cpp, 3.cpp, 4.cpp.

Pliki te umieść w archiwum skompresowanym .zip (zaznacz pliki z kodem źródłowym>PPM>Wyślij do>Archiwum skompresowane). Nadaj temu archiwum nazwę klasaGrupa_nazwisko (np. 3F1_Jarecki). Archiwum wyślij na FTP do folderu 3G.