Część XX C++ w

Elementy poza tablicą

Do tej pory wpisywaliśmy do tablicy tylko tyle elementów, ile dla niej zadeklarowaliśmy. Czy nie zastanawia nas, co się stanie, jeśli na przykład spróbujemy do tablicy pięcioelementowej wpisać wartość szóstego elementu?

Ćwiczenie 1

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko70**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

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

using namespace std;

int main(int argc, char *argv[])
{
    int tablica[5] = {5,4,3,2,1};
    cout << "Rozmiar tablicy: " << sizeof(tablica) << endl;

    for (int i=1; i<5; i++) cout << i << " -> " << tablica[i] << endl;

    cout << endl << "5 -> " << tablica[5] << endl;

    tablica[5] = 20;
    cout << endl << "5 -> " << tablica[5] << endl;

    cout << "Rozmiar tablicy: " << sizeof(tablica) << endl;


    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Na początku definiujemy zmienną tablicową **tablica** o rozmiarze pięciu elementów i inicjujemy ją dowolnymi wartościami.

Wyświetlimy teraz rozmiar naszej tablicy.

Funkcja **sizeof ()** zwraca ilość bajtów zajmowanych w pamięci przez zmienną o przekazanej do niej nazwie. Tak więc na ekranie widzimy, że zmienna **tablica** zajmuje w pamięci naszego komputera 20 bajtów (pięć elementów typu **int** - każdy po 4 bajty).

Za pomocą pętli wyświetlamy elementy tablicy

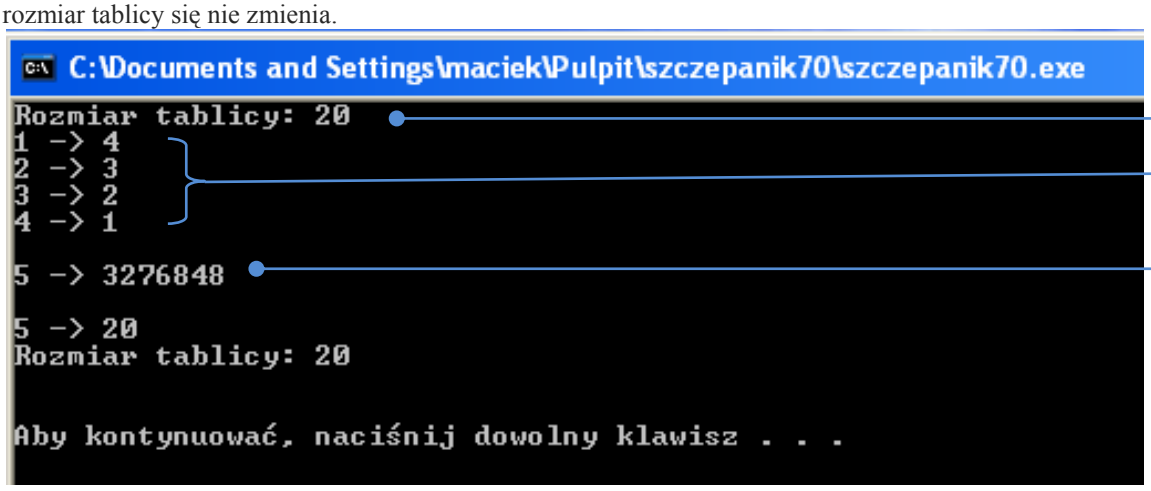
Spróbujmy teraz wyświetlić wartość szóstego elementu (o indeksie 5).

Łatwo przekonamy się, że jest to możliwe - na ekranie pojawia się dowolna liczba.

Jej wartość będzie odpowiadała wartości komórki pamięci, która znajduje się tuż za naszą tablicą.

Wiemy, że możemy wyświetlić wartość elementu tablicy, który tak naprawdę się w niej nie znajduje (leży po prostu w pamięci komputera tuż za tablicą). Czy możemy coś zapisać do tego elementu? Warto spróbować. Szybko okaże się, że jest to możliwe. Po ponownym wyświetleniu tego elementu widzimy, że wartość została normalnie przypisana.

Na koniec sprawdzamy jeszcze rozmiar naszej tablicy - widzimy, że mimo przypisania wartości dodatkowemu elementowi rozmiar tablicy się nie zmienia.



Bezwymiarowa tablica wielowymiarowa

Pamiętajmy, że kompilator może samodzielnie przydzielić tylko jeden wymiar tablicy. Tak więc zapis nie zostanie poprawnie skompilowany.

```
int tablica[][3] = {{1, 2, 3}, {4, 5, 6}};
```

```
int tablica[][] = {{1, 2, 3}, {4, 5, 6}};
```

Bez wymiaru możemy pozostawić tylko jeden i do tego wyłącznie pierwszy wymiar tablicy. Zapis jest więc jak najbardziej poprawny.

Tablica ma zawsze taki wymiar (zostaje jej przydzielone tyle miejsca w pamięci), jaki ustalimy podczas jej deklaracji. Oczywiście można czytać i zapisywać wartości w elementach o indeksach większych niż ostatni indeks tablicy, ale nie należy tego robić. Dlaczego? Ponieważ możemy nadpisać obszar pamięci nienależący do naszej tablicy, ale przydzielony dla innej zmiennej, czy też dla innej aplikacji.

Tablica znakowa

Do tej pory posługiwaliśmy się najczęściej tablicami typu **int**. Oczywiście nasza tablica może być innego typu (może na przykład przechowywać liczby zmiennoprzecinkowe) - dla działania takiej tablicy nie ma to żadnego znaczenia. Jedynym wyjątkiem jest typ znakowy **char**, dzięki któremu możemy tworzyć dosyć specyficzne tablice. Wyobraźmy sobie bowiem, że jako kolejne elementy tablicy zapisujemy znaki: K, o, m, p, u, t, e, r. Jeśli wyświetlimy później w pętli kolejne elementy tablicy, na ekranie zobaczymy napis Komputer. Tablica typu znakowego umożliwia nam więc przechowywanie ciągów znakowych, czyli napisów..

Ćwiczenie nr 2

1.Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko71**
2.Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

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

using namespace std;

int main(int argc, char *argv[])
{
    char napis[] = {'Z','n','a','c','z','k','i',' ',' ','R','O','M','K','A','\0'};

    for (int i=0; napis[i]!='\0'; i++) cout << napis[i];

    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Deklarujemy tablicę typu znakowego o nazwie napis i inicjujemy ją wartościami

Jako ostatnią wartość umieszczamy znak zerowy - symbol \0

Rozmiar tablicy **napis** kompilator automatycznie ustawia na 15 elementów.

W pętli typu for wyświetlamy kolejne elementy tablicy **napis** Zwróćmy uwagę na warunek przejścia do kolejnego przebiegu pętli. Można go odczytać następująco: Wykonuj pętlę, dopóki wartość kolejnego elementu tablicy **napis** nie będzie symbolem \0.

3.Z menu Uruchom wybieramy Kompiluj i uruchom.

Łańcuch znakowy

Oczywiście posługiwanie się łańcuchami znakowymi w przedstawiony przed chwilą sposób byłoby bardzo niewygodne (a szczególnie pamiętanie o znaku zerowym). Na szczęście istnieją prostsze metody zapisania łańcucha znakowego.

Ćwiczenie nr 3

Zobaczmy jak zadeklarować tablicę dwuwymiarową i jak z niej korzystać

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko72**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

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

using namespace std;

int main(int argc, char *argv[])
{
    char napis[]="Dzien dobry";
    char imie[10];

    cout << "Podaj swoje imie: ";
    cin >> imie;

    cout << napis << " " << imie;

    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Na początku deklarujemy tablicę bezwymiarową (oczywiście typu znakowego) i inicjujemy ją łańcuchem znaków zapisanym w cudzysłowach.

Taki zapis jest znacznie wygodniejszy niż wpisywanie jako elementów tablicy kolejnych znaków.

Tablicy (możemy ją zwać łańcuchem znaków) o nazwie **napis** zostaje przydzielony rozmiar 12 (11 znaków plus znak zerowy na końcu).

Tworzymy jeszcze jedną tablicę znakową - o nazwie **imie** i rozmiarze 10 znaków.

Zapisujemy w niej imię, które użytkownik wpisuje z klawiatury.

Teraz możemy wyświetlić oba łańcuchy znaków.

W tym celu podajemy nazwę zmiennej tablicowej (bez nawiasów kwadratowych).

3. Z menu Uruchom wybieramy Kompiluj i uruchom.

W ostatnim przykładzie zadeklarowaliśmy rozmiar zmiennej tablicowej **imie** na 10 znaków (a właściwie 9 znaków i znak zerowy). Co się stanie, jeśli użytkownik poda wyraz znacznie dłuższy od przewidzianego przez nas (na przykład taki **wertyuioplkjhgfdsazxcvbnmwertyuiodfgh**)? Dziesięć pierwszych znaków tego napisu zostanie zapisanych w pamięci zarezerwowanej dla tablicy **imie**. Pozostałe znaki zapiszą się w kolejnych komórkach pamięci, nadpisując znajdujące się tam dane. W tym wypadku część znaków została wpisana w obszarze pamięci przeznaczonym dla tablicy **napis**. Dlatego zamiast ciągu „Dzień Dobry" na ekranie pojawia się fragment podanego przez użytkownika napisu . Pamiętajmy więc, aby deklarować rozmiar tablicy na tyle duży, aby pomieścił planowany łańcuch znaków. Nie warto jednak tworzyć bardzo dużych tablic - niepotrzebnie zajmują miejsce w pamięci.

Lepsze łańcuchy znaków

Łańcuchy znaków w postaci tablicy są niezbyt wygodne. Dlatego stworzono specjalny typ danych służący do reprezentacji napisów. Typ ten nazwano string – spotkaliśmy się z nim przy okazji omawiania stałych.

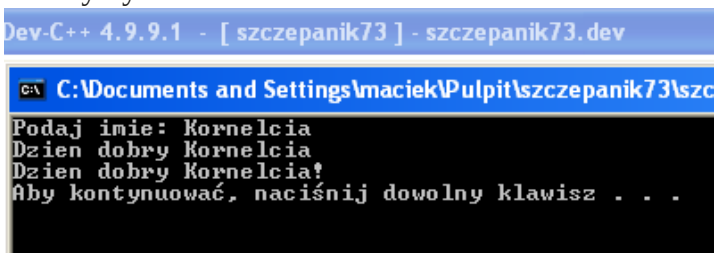
String ma wiele zalet w porównaniu z tablicą znakową. Warto wymienić te najważniejsze:

- ►używając typu string, nie musimy martwić się o przydział pamięci (zadeklarowanie tablicy o odpowiedniej wielkości),
- ► zmienna typu string zajmuje mniej miejsca niż tablica znaków, której rozmiar powinien być na tyle duży, aby zmieścił się w niej każdy napis, jaki przewidujemy w niej zapisać,

► wykorzystując string, mamy możliwość skorzystania z wielu pomocnych funkcji - wyszukujących, łączenia różnych napisów, porównywania dwóch napisów ze sobą.

Definicja typu danych **string** znajduje się w jednym z plików dołączanych zawsze na początku naszego kodu za pomocą polecenia **#include <iostream>.**

Aby pokazać działanie typu string, wykonajmy ćwiczenie po skompilowaniu którego na ekranie zobaczymy



Ćwiczenie 4

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko73**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

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

using namespace std;

int main(int argc, char *argv[])
{
    string napis = "Dzien dobry ";
    string imie;

    cout << "Podaj imie: ";
    cin >> imie;

    cout << napis << imie << endl;

    napis += imie + "!";
    cout << napis << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Warto zwrócić uwagę na zapis , który pokazuje możliwość łączenia (za pomocą zwykłego operatora sumowania) wielu zmiennych typu string w jeden łańcuch znakowy

3.Z menu Uruchom wybieramy Kompiluj i uruchom

Wszystkie pliki z nazwiskiem i kolejnym numerem umieszczamy w swoim folderze nazwiskocplusplus na serwerze.