Scenariusz zajęć nr 2

Temat: Plan lekcji

Cele:

W trakcie zajęć uczeń zapozna się z następującymi pojęciami:

- Podstawy użytkowania środowiska programistycznego (IDE) dla języka C++ (Code::Blocks lub Dev-Cpp).
- Wczytywanie danych liczbowych (`cin >>`),
- Wypisywanie danych liczbowych (`cout << ... << endl`),
- Deklarowanie zmiennych o typie całkowitym (`int`), • Deklarowanie i używanie tablicy o wartościach całkowitych,
- Używanie instrukcji pętli `for`,
- Używanie instrukcji warunkowej `if ...`,
- Używanie warunków logicznych z operatorami porównania `<=` oraz `==`,
- Użycie operatora logicznego koniunkcji `&&`.
- Sumowanie zawartości tablicy (opcjonalne użycie operatora `+=`),

Wstęp:

Uruchomienie środowiska programistycznego, zapoznanie się z edytorem kodu źródłowego i sposobem kompilacji programu (kontynuacja).

Dla nauczyciela:

W razie braku zainstalowanego środowiska można skorzystać z dowolnego środowiska dostępnego w sieci Internet, na przykład `cpp.sh`.

Przebieg zajęć:

Zapoznanie się z treścią zadania

Treść zadania jest dołączona do scenariusza zajęć w formie pliku `cpp_02-plan.pdf`.

Dla nauczyciela:

Tematem zajęć jest zadanie *Plan lekcji* dostępne na serwisie szkopul.edu.pl, wzorowane na zadaniu *Alena's Schedule* (586A) z serwisu codeforces.com. Zadanie można rozwiązać w dowolnym języku programowania, a jego stopień trudności należy ocenić jako właściwy dla uczestników początkowego kursu programowania.

Wskazówki do zadania

Halinka musi bezwarunkowo pozostawać na uczelni podczas każdej godziny zajęć oznaczonej liczbą 1. Ponadto jeśli trafi się pojedyncza odosobniona wolna godzina taka, że przed nią i po niej są obowiązkowe zajęcia, to wtedy również musi pozostać na uczelni, bo nie opłaca się jej wracać do domu i przyjeżdżać znowu. Na przykład w poniższym przykładzie podkreślone godziny

Projekt Centrum Mistrzostwa Informatycznego współfinansowany jest ze środków Unii Europejskiej z Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014 - 2020.

Halinka spędza na uczelni:

1 0 1 1 1 0 0 1 0 1 0

Zatem należy wczytać dane do tablicy (na pozycje od 1 do *n*), a następnie przejrzeć tę tablicę na pozycjach *i* = 2, 3, . . . , *n* − 1 i jeśli na pozycjach *i* − 1 oraz *i* + 1 znajdują się jedynki, wtedy na pozycji *i* trzeba wpisać 1.

Na koniec należy zsumować zawartość tej tablicy – to będzie wynik zadania.

Kod przykładowego programu w C++

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

int main()
{
  int n;
  cin >> n;
  int a[101];
  for(int i = 1; i <= n; i++)
    cin >> a[i];
  for(int i = 2; i <= n - 1; i++)
    if(a[i - 1] && a[i + 1])
      a[i] = 1;
  int s = 0;
  for(int i = 1; i <= n; i++)
    s += a[i];
  cout << s << endl;
  return 0;
}
```

Podsumowanie i dodatkowe uwagi:

Rozmiar deklarowanej tablicy musi być wartością stałą. Nie jest zatem dopuszczalna poniższa konstrukcja:

```
int n;
cin >> n;
int a[n];
```

Kompilator musi znać rozmiar tablicy w momencie jej deklaracji – taki jest wymóg standardu języka C++ (źródło: Bjarne Stroustrup *Język C++*). Dostępne na rynku kompilatory nie sygnalizują w powyższym przypadku błędu, ale program może działać nie do końca poprawnie. Dlatego też w takich zadaniach deklaruje się tablice o maksymalnym rozmiarze określonym w specyfikacji danych wejściowych w treści zadania. Nie ma bowiem żadnej premii za oszczędzanie pamięci RAM – wystarczy zmieścić się w zadanym limicie, który najczęściej wynosi 256 MB.

Projekt Centrum Mistrzostwa Informatycznego współfinansowany jest ze środków Unii Europejskiej z Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014 - 2020.

Dlaczego jednak zadeklarowaliśmy tablicę o rozmiarze 101, a nie 100? Otóż w większości zadań algorytmicznych dane numerowane są począwszy od indeksu 1, na przykład: a_1, a_2, a_3, . . . , a_n (tak jest też w tym zadaniu) – natomiast elementy struktur w typowych językach programowania (w tym C++ i Pythonie) numerowane są od indeksu 0. Zatem, jeśli chcemy używać elementu struktury o indeksie n (w tym zadaniu maksymalnie 100), rozmiar struktury musi być o 1 większy, zaś elementu zerowego po prostu się nie używa. Podczas sumowania zawartości poprawionej tablicy a[ ] można użyć standardowej instrukcji:

```
s = s + a[i];
```

albo jej skróconej wersji:

```
s += a[i];
```

Warto podkreślić różnicę pomiędzy instrukcją podstawienia a równością w sensie algebraicznym, co jest zwłaszcza widoczne przy instrukcji `s = s + a[i]`.

Biuro Projektu Partnera Wiodącego
ul. B. Stefanowskiego 18/22, lokal 14
90-924 Łódź
tel.: (42) 631-28-86

Projekt Centrum Mistrzostwa Informatycznego współfinansowany jest ze środków Unii Europejskiej z Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014 - 2020.