Języki programowania – lista 6 Wrocław, 16.11.2012

1. Napisz funkcję o nazwie **print(const vector<int>& v)** drukującą zawartość wektora liczb typu **int** w strumieniu **cout**. Niech przyjmuje jako argument wektor.

2. Napisz dwie funkcje odwracające kolejność elementów w wektorze **vector<int>**.
   a) Niech pierwsza z tych funkcji tworzy nowy wektor zawierający elementy w odwróconym porządku, oryginalny pozostawiając bez zmian. Jej prototyp może wyglądać tak:
   vector<int> wektor_odwrocony(const vector<int>& v)
   b) Druga funkcja niech odwraca porządek elementów w oryginalnym wektorze bez pomocy innego wektora. *Podpowiedź: użyj funkcji* ***swap()***. Jej prototyp może wyglądać tak:
   void odwroc_kolejnosc(vector<int>& v)
   Zwróć uwagę, że funkcje umyślnie dostały inne nazwy.
   Napisz program demonstrujący działanie funkcji. Do wyświetlania wektorów użyj funkcji **print()**.

3. Utwórz wektor liczb ciągu Fibonocciego i wydrukuj za pomocą funckji **print()**. Napisz funkcję **fibonacci(x,y,v,n)** zwracającą wektor, gdzie **x** i **y** to liczby typu **int**, **v** jest pustym wektorem **vector<int>**, a **n** oznacza liczbę elementów, które mają być wstawione do **v**. **v[0]** będzie **x**, a **v[1]** będzie **y**. Funkcja **fibonacci()** powinna tworzyć ciągi Fibonacciego, zaczynając od argumentów **x**, **y**.

4. Białko składa się z sekwencji 20 aminokwasów, które można reprezentować literami alfabetu. Każdy aminokwas ma pewne własności, np. ładunek elektryczny netto. Większość aminokwasów ma ładunek neutralny, niektóre mają ujemny, inne dodatni. Napisz program, który dla podanej sekwencji aminokwasów wyświetla profil ładunku oraz sumaryczny ładunek netto. Np.

```
Podaj sekwencje aminokwasów:
       FGGLIRDVKRRY
Profil .....+-.+++.
Ladunek netto: +3e
```

   Program powinien składać się z następujących funkcji: i) funkcja zwracająca ładunek netto dla podanej litery reprezentującej aminokwas, ii) funkcja zwracająca ładunek elektryczny netto dla zadanej sekwencji aminokwasów, iii) funkcja wyświetlająca profil ładunku dla zadanej sekwencji aminokwasów.
   Pamiętaj o raportowaniu wyjątków.

Dla ambitnych

5. Rozbuduj program z zadania 4 o możliwość wyszukiwania fragmentów sekwencji o zadanej długości **n**, których bezwzględny ładunek jest większy równy **q**. W tym celu napisz funkcję przyjmującą jako argumenty sekwencję aminokwasów, **n** oraz **q**, która zwraca wektor z pozycjami znalezionych fragmentów w oryginalnej sekwencji. W swoim programie wywołaj tę funkcję, a następnie wyświetl fragmenty, np.

```
Podaj sekwencje aminokwasów:
       FGGLIRDVKRRY
Profil .....+-.+++.
Ladunek netto: +3e
Dlugosc fragmentu: 5
Ladunek bezwzgledny >=: 2
Znaleziono fragmenty:
1. poz. 6: RDVKR
   q=2e    +-.++
2. poz. 7: DVKRR
   q=2e    -.+++
3. poz. 8: VKRRY
    q=3e    .+++.
```

6. Napisz program wczytujący imiona do wektora **imie** typu **vector<string>**. Następnie wczytaj wiek tych osób do wektora **wiek** typu **vector<int>**. Posortuj imiona (**sort(imie.begin(), imie.end())**) i wydrukuj pary imię-wiek. Trudność zadania polega na tym, by w wektorze **wiek** uzyskać taki sam porządek jak w wektorze **imie**. *Wskazówka: przed sortowanie zrób kopię wektora imion.* Co jeśli imię nie będzie unikalne? Pamiętaj o raportowaniu wyjątków.

Powodzenia! Witold Dyrka i zespół