Programowanie II R

Zadania – seria 6.

STL: kontener `std::vector`.

Zadanie 1. `exactsum` – Ciąg liczb o zadanej sumie.

Napisz funkcję

```
std::vector<int> ExactSum(int k, const std::vector<int> &numbers)
```

która przyjmuje jako argumenty liczbę całkowitą `k` oraz stałą referencję wektora liczb całkowitych `numbers`. Funkcja powinna odszukać w tym wektorze ciąg następujących po sobie liczb o sumie równej `k` i zwrócić wektor zbudowany z wyrazów tego ciągu. Gdy ciągów takich w wektorze `numbers` jest kilka, zwracany wektor powinien zawierać wyrazy „pierwszego" z nich, czyli tego, którego początkowy wyraz leży najbliżej pierwszego elementu wektora `numbers`; jeśli zaś nie ma ani jednego takiego ciągu, funkcja powinna zwrócić pusty wektor.

Korzystając z tej funkcji napisz program `exactsum`, przyjmujący jako argument wywołania liczbę całkowitą. Program powinien wczytać ze standardowego wejścia ciąg liczb całkowitych do napotkania znaku końca pliku, a następnie znaleźć w nim podciąg o sumie wyrazów równej liczbie całkowitej przekazanej jako argument wywołania i wypisać ten podciąg na standardowe wyjście. Gdy podciągów takich jest kilka, program powinien wypisać „pierwszy" z nich, czyli ten, którego początkowy wyraz leży najbliżej początkowego wyrazu wyjściowego ciągu; jeśli zaś nie ma ani jednego takiego podciągu, program powinien wypisać stosowny komunikat.

Zadanie 2. `selectionsort` – Sortowanie przez wybieranie.

Sortowanie przez wybieranie (ang. *selection sort*) to jeden z najprostszych znanych algorytmów sortowania. W celu posortowania pewnej kolekcji elementów w kolejności rosnącej metodą przez wybieranie należy najpierw odszukać najmniejszy element tej kolekcji i zamienić go miejscami z jej pierwszym elementem – element najmniejszy znajdzie się wówczas na swojej docelowej pozycji. Czynności te należy następnie powtórzyć dla pozostałych elementów należących do sortowanej kolekcji: szukamy najmniejszego z nich i zamieniamy go miejscami z elementem na drugiej pozycji – w ten sposób dwa pierwsze elementy kolekcji będą na właściwych pozycjach. Procedurę tę należy powtarzać tak długo, aż wszystkie elementy kolekcji znajdą się na swoich miejscach (czyli – innymi słowy – aż zbiór elementów wciąż wymagających posortowania stanie się jednoelementowy).

Napisz funkcję

```
void SelectionSort(std::vector<int> &data)
```

która przyjmuje jako argument referencję wektora liczb całkowitych, a następnie sortuje ten wektor rosnąco, posługując się algorytmem sortowania przez wybieranie.

Korzystając z tej funkcji napisz program `selectionsort`, który wczytuje ze standardowego wejścia liczby całkowite do napotkania znaku końca pliku, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej.

Zadanie 3. `lagrange` – Interpolacja wielomianowa.

Rozważmy zbiór $n+1$ punktów płaszczyzny

$$\{(x_k,\, y_k)\}_{k=0}^{n} = \{(x_0,\, y_0)\,,\, (x_1,\, y_1)\,,\, \ldots,\, (x_n,\, y_n)\}\,, \qquad \forall\, 0 \leqslant k \leqslant n :\, (x_k,\, y_k) \in \mathbb{R}^2.$$

Wielomian interpolacyjny Lagrange'a, określony wzorem

$$w(x) \;=\; \sum_{i=0}^{n} y_i \prod_{j=0 \wedge j \neq i}^{n} \frac{x - x_j}{x_i - x_j},$$

jest wielomianem n–tego stopnia spełniającym

$$\forall\, 0 \leqslant k \leqslant n :\, w\,(x_k) = y_k.$$

Punkty $\{(x_k,\, y_k)\}_{k=0}^{n}$ nazywa się *węzłami* tego wielomianu.

Napisz funkcję

```
double Lagrange(const std::vector<double> &px, const std::vector<double> &py,
    double x)
```

która zwróci wartość wielomianu interpolacyjnego Lagrange'a dla argumentu `x`. Wektory `px` i `px` zawierają, odpowiednio, odcięte (współrzędne x) i rzędne (współrzędne y) węzłów wielomianu. Funkcja nie powinna tworzyć żadnych dodatkowych kolekcji.

Korzystając z tej funkcji, napisz program `lagrange`, który wczytuje ze standardowego wejścia wartość `x` oraz ciąg liczb zmiennoprzecinkowych aż do napotkania znaku końca pliku i traktuje je jako współrzędne punktów, zapisując ich wartości odpowiednio do wektorów `px` i `py`. Program powinien wyznaczać i wypisywać na standardowe wyjście wartość zwracaną przez funkcję `Lagrange`.

Opracowanie: Bartłomiej Zglinicki.