

Podstawy programowania, poniedziałek 11.04.2016, 8-10

Zadanie 5

Urząd statystyczny postanowił, w ramach okresowego badania populacji, zmierzyć wzrost pewnej reprezentatywnej grupy studentów. Do opracowania wyników badania niezbędne jest: policzenie średniego wzrostu studentów z badanej próbki, odchylenia standardowego wartości średniej, oraz posortowanie danych.

Plik z danymi ma następujący format:

```
liczba_studentów
wzrost1
wzrost2
...
```

Do wykonania

- Stworzyć (dynamicznie) tablicę liczb rzeczywistych (float) o rozmiarze danym liczbą zbadanych studentów i wczytać do niej z zewnętrznego pliku dane.

- Wypisać na ekran tablicę danych – funkcja zewnętrzna

```
void wypisz(const float* tab, int n). (1 p.)
```

- Obliczyć i wypisać na ekran wartość średnią wczytanej próbki danych – funkcja wewnętrzna

```
float srednia(const float* tab, int n). (1 p.)
```

- Obliczyć i wypisać na ekran odchylenie standardowe wartości średniej (jak policzyć odchylenie standardowe – patrz **Uwaga 1** poniżej) – funkcja zewnętrzna

```
float odchStd(const float* tab, int n). (1 p.)
```

- Zaimplementować funkcję sortującą dane za pomocą algorytmu sortowania przez wstawianie, tzw. Insert Sort (patrz **Uwaga 2** poniżej) – funkcja wewnętrzna

```
void sortuj(float* tab, int n). (1.5 p.)
```

- Program powinien działać w pętli while, a każda z powyższych opcji powinna być realizowana za pomocą odpowiedniej opcji używając switch-case. (1.5 p.)

- Plik z danymi należy ściągnąć z:

<http://www.if.pw.edu.pl/~lgraczyk/PP2016/lab05/wzrost.txt>

Uwaga 1

Odchylenie standardowe wartości średniej:

$$u(x) = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Aby policzyć pierwiastek z liczby używamy funkcji `sqrt` lub jej wariantu `sqrtf` dla typu `float`:

```
double sqrt (double x);
```

```
float sqrtf (float x);
```

Należy dołączyć bibliotekę: `#include <math.h>`

Uwaga 2

Schemat działania algorytmu sortowania przez wstawianie (za *Wikipedia*):

1. Utwórz zbiór elementów posortowanych i przenieś do niego dowolny element ze zbioru nieposortowanego.
2. Weź dowolny element ze zbioru nieposortowanego.
3. Wyciągnięty element porównuj z kolejnymi elementami zbioru posortowanego póki nie napotkasz elementu równego lub elementu większego (jeśli chcemy otrzymać ciąg niemalejący) lub nie znajdziemy się na początku/końcu zbioru uporządkowanego.
4. Wyciągnięty element wstaw w miejsce gdzie skończyłeś porównywać.
5. Jeśli zbiór elementów nieuporządkowanych jest niepusty wróć do punkt 2.

Pseudokod *sortowania przez wstawianie* sortujący tablicę A:

```
for i = 1 to length(A) - 1
  x = A[i]
  j = i
  while j > 0 and A[j-1] > x
    A[j] = A[j-1]
    j = j - 1
  end while
  A[j] = x
end for
```