

Kod źródłowy rozwiązania każdego z zadań zapisz z oddzielnym pliku *.cpp o nazwie INDEKS_NR ZADANIA. Programy nie kompilujące się otrzymują **zero** punktów. Można korzystać z Internetu, jednak zabronione są wszelkie formy kontaktowania się. Powodzenia!

Zadanie 1. (10 pkt) Napisz klasę *Naukowiec* zawierającą nazwisko, liczbę publikacji oraz cytowań naukowca. Klasa powinna zapewniać następujące operacje:

- konstruktor trójargumentowy inicjalizujący nazwisko, liczbę publikacji oraz cytowań,
- funkcję składową *indeks* zwracającą średnią liczbę cytowań na jedną publikację,
- funkcję składową *publikuj* zwiększającą liczbę publikacji o zadaną wartość,
- operator preinkrementacji *++* zwiększający o 1 liczbę cytowań naukowca,
- operator *<* porównujący naukowców. Lepszy jest naukowiec z większą liczbą cytowań, a jeżeli są takie same, to ten dla którego funkcja *indeks* zwraca większą wartość.
- operator *<<* wypisywania danych naukowca do strumienia typu *ostream* oraz operator *>>* wczytywania danych naukowca ze strumienia *istream*.

PRZYKŁAD UŻYCIA KLASY:

```
int main() {
    Naukowiec Bogdan("Bogdan",50,300), Czeslaw;
    cin >> Czeslaw;
    cout << Czeslaw.indeks();
    Bogdan.publikuj(3);
    if(Bogdan < ++Czeslaw)
        cout << Czeslaw << endl;
    else
        cout << Bogdan << endl;
}
```

Zadanie 2. (10 pkt) Sprawdzanie kolokwium z programowania to bardzo ciężka praca. Wyniki kolokwium znajdują się w pliku tekstowym, zapisanym w następującym formacie: w każdej linii znajduje się jedno imię i jedno nazwisko studenta, a następnie liczby punktów za kolejne zadania. Liczba zadań ani osób nie jest z góry znana, wiadomo jednak, że liczba zadań jest dla wszystkich studentów taka sama oraz plik z danymi nie jest pusty. Napisz program, który na podstawie tego pliku obliczy i wypisze do pliku *wyniki.txt* całkowitą liczbę punktów zdobytych przez każdego studenta oraz średnią liczbę punktów z każdego zadania. Nazwa pliku zawierającego dane wejściowe powinna być podawana jako argument wywołania programu. *Wskazówka:* możesz użyć pojemnika *<vector>* z biblioteki STL oraz metod *clear* i *str* dla obiektu strumienia *istream*. PRZYKŁAD WEJŚCIA I WYJŚCIA:

PLIK WEJŚCIOWY: (dane.txt)

Jakub Wedrowycz 1.5 0 0.5
Semen Korczaszko 5.5 10 3.5
Jozef Paczenko 4 5 6.5

PLIK WYJŚCIOWY: (wyniki.txt)

Jakub Wedrowycz 2
Semen Korczaszko 20
Jozef Paczenko 15.5
z1 4 z2 5 z3 3.5

Zadanie 3. (10 pkt) Napisz klasę `Statystyka` obliczająca średnią, odchylenie standardowe oraz medianę ciągu liczb o dowolnej długości. Klasa powinna udostępniać funkcje:

- `void dodaj (double x)` — uwzględnia w obliczeniach nową liczbę x ,
- `double srednia()` — przekazuje średnią arytmetyczną dotychczas zapamiętanych liczb,
- `double sigma()` — przekazuje odchylenie standardowe dotychczas zapamiętanych liczb,
- `double mediana()` — przekazuje wartość mediany dotychczas zapamiętanych liczb,
- `void zeruj()` — usuwa zapamiętane liczby (przywraca początkowy zbiór pusty).

Napisz program, który losuje N liczb rzeczywistych z rozkładu płaskiego w przedziale $[-k, k]$, gdzie $N, k \in \mathbb{N}$ są argumentami wywołania programu. Program powinien zwracać na wyjściu średnią, odchylenie standardowe oraz medianę wylosowanych liczb. *Wskazówka:* możesz użyć pojemnika `<vector>` z biblioteki STL oraz funkcji `atoi` z biblioteki `<cstdlib>`.
PRZYKŁAD UŻYCIA KLASY:

```
int main() {  
    Statystyka stat;  
    for (int i = 0; i<10; i++)  
        stat.dodaj(i);  
    cout << stat.srednia() << ' ' << stat.sigma() << ' ' << stat.mediana();  
    stat.zeruj();  
}
```