

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)** Dany jest ciąg liczb całkowitych zdefiniowany rekurencyjnie jako:

$$T_0 = 1, \quad T_1 = 1, \quad T_2 = 2, \quad T_n = T_{n-1} + T_{n-2} + T_{n-3}.$$

Napisz funkcję **rekurencyjną** służącą do obliczania kolejnych wyrazów ciągu. Następnie zastosuj ją w programie, który wypisuje do pliku tekstowego wyniki.txt pierwszych  $n$  wyrazów ciągu  $T_n$ . Liczba naturalna  $n$  powinna być podawana jako argument wywołania programu. *Wskazówka*: możesz użyć funkcji `atoi` z biblioteki `<cstdlib>`. PRZYKŁAD:

```
Podana liczba naturalna n: 3
Zawartość pliku tekstowego: 1 1 2
```

```
Podana liczba naturalna n: 6
Zawartość pliku tekstowego: 1 1 2 4 7 13
```

**Zadanie 2. (10 pkt)** Napisz klasę bazową `Funkcja` reprezentującą dziedzinę  $[x_D, x_U]$  dowolnej funkcji. Klasa powinna posiadać pola publiczne przechowujące wartości  $x_D$  i  $x_U$  ( $-10^3 \leq x_D, x_U \leq 10^3$ ) oraz konstruktor dwuargumentowy umożliwiający ich inicjalizację.

Następnie napisz klasy pochodne `Liniowa` oraz `Kwadratowa` dziedziczące publicznie po klasie `Funkcja`, reprezentujące **miejsca zerowe** funkcji liniowej ( $y = Ax + B$ ) oraz kwadratowej ( $y = Ax^2 + Bx + C$ ). Każda z nich powinna posiadać pola przechowujące parametry i miejsca zerowe funkcji oraz konstruktor wywołujący metodę `void mZerowe()`, która znajduje **miejsca zerowe** funkcji. W przypadku, gdy miejsca zerowe funkcji nie istnieje lub znajduje się poza dziedziną, przypisz mu wartość  $\text{Inf} = 10^6 - 1$ .

PRZYKŁAD UŻYCIA KLAS:

```
int main() {
    Liniowa fun1(-1.5, 1, 5, 2.5); // [-1.5, 1], A=5, B=2.5
    cout << fun1.m0 << endl; // m0 = -0.5
    Kwadratowa fun2(-2.5, 5, 2, -1, -28); // [-2.5, 5], A=2, B=-1, C=-28
    cout << fun2.m1 << '\t' << fun2.m2 << endl; // m1 = Inf, m2 = 3.5
}
```

**Zadanie 3. (10 pkt)** Korzystając z klasy `Naukowiec` (plik nagłówkowy `Naukowiec.h` oraz źródłowy `Naukowiec.cpp`) napisz klasę `Naukowcy` przechowującą informacje o grupie naukowców. Klasa powinna zapewniać następujące operacje:

- konstruktor jednoargumentowy inicjalizujący nazwę grupy i liczbę członków równą 0,
- operator `+=` dodający do grupy podanego naukowca (na dowolnej pozycji),
- funkcję `sredlIndex` zwracającą średnią arytmetyczną indeksów wszystkich naukowców,
- operator predekrementacji `--` i preinkrementacji `++` znajdujące najgorszego i najlepszego naukowca w danej grupie (*patrz*: operator `<` w klasie `NAUKOWIEC`),
- operator `<` porównujący grupy naukowców. Lepsza jest grupa z większym średnim indeksem, a jeżeli jest taki sam, to ta zawierająca mniejszą liczbę naukowców.
- operator `+` tworzący z dwóch oddzielnych grupy naukowców jedną *supergrupę*,
- operator `<<` wypisywania danych o grupie naukowców (nazwa grupy i liczebność) do strumienia typu `ostream`. Następnie powinny zostać wypisane dane wszystkich naukowców (imię, publikacje i cytowania).

Kod projektu podziel na pliki nagłówkowe i źródłowe (osobne dla dostarczonej klasy `Naukowiec` i klasy `Naukowcy`). Przykłady użycia klas znajdują się w plikach `testNaukowiec.cpp` oraz `testNaukowcy.cpp`. *Wskazówka*: możesz użyć pojemnika `<vector>` z biblioteki STL.