

Laboratorium 1

1. Scena koncertowa

Ekipa realizująca koncert muzyczny ma za zadanie oświetlać scenę za pomocą 10 reflektorów ustawionych w różnych punktach sali. Każdy z reflektorów emituje wiązkę światła RGB, o indywidualnych proporcjach barw R, G i B. Efekt sumaryczny uzyskuje się przez uśrednienie tych wiązek.

Napisz program wspomagający "wyznaczanie" światła padającego na centralny punkt sceny, oświetlany przez wszystkie reflektory. Dane zawarte są w pliku `swiatla.txt`. Pierwszy rekord pliku zawiera liczbę naturalną n – liczbę zestawów danych następujących kolejno za n . Pojedynczy zestaw składa się z 10 rekordów, zawierających po trzy liczby całkowite, reprezentujące ilość światła barwy, odpowiednio R, G i B ($0 \leq R, G, B \leq 255$), oddzielone spacjami (co najmniej jedną).

Dla każdego zestawu danych, wypisz trzy **liczby całkowite** (R, G i B) oznaczające ilość odpowiedniej barwy w świetle padającym na centralny punkt sceny. Podaj wynik na standardowe wyjście, z uwzględnieniem zaokrąglenia matematycznego do najbliższej liczby całkowitej, tzn. cyfry od 1 do 4 należy zaokrąglić w dół, natomiast od 5 do 9 w górę.

Przykład

Dane w pliku `swiatla.txt`:

```
1
0 0 255
0 255 0
0 255 255
255 0 0
255 0 255
255 255 0
255 255 255
0 0 0
206 4 105
200 0 100
```

Dane wyjściowe:

```
143    2 123
```

2. Równanie kwadratowe

Daną są współczynniki rzeczywiste A, B i C równania kwadratowego $Ax^2 + Bx + C = 0$, oraz dodatkowo – dokładność `eps`, wyrażona liczbą rzeczywistą stałoprzecinkową o wartości 10^{-n} , gdzie n jest liczbą całkowitą dodatnią. Podaj z dokładnością `eps`, w formacie stałoprzecinkowym, wartość większego z pierwiastków tego równania. W przypadku braku pierwiastków rzeczywistych ($\Delta < 0$), poinformuj o tym stosownym komunikatem.

Przykład

Dane wejściowe

```
1 2 0.5 0.01 //A B C eps
```

Dane wyjściowe

```
-0.29
```

3. Słowa i podśłowa

Sprawdź, czy ciąg znaków zadany w pierwszym wierszu standardowego wejścia jest podśłowem ciągu zadanego w drugim wierszu tego wejścia.

Przykład1

Dane wejściowe

a ma kota

Ala ma kota, ale nie ma psa

Dane wyjściowe

tak

Przykład2

Dane wejściowe

A, ma kota!

Ala ma kota, ale nie ma psa

Dane wyjściowe

nie

Zadanie obowiązkowe

4. Porządkowanie słów

W pliku tekstowym 'słowa.txt' zapisano różnej długości słowa. Słowa są zbudowane z małych i wielkich liter alfabetu łacińskiego, a ich długość nie przekracza 9. Wpisz wszystkie słowa z pliku do wektora list TABL, zawierającego w i-tej pozycji listę słów o długości i. Na pozycji 0 tego wektora powinien znajdować się wskaźnik NULL. Uporządkuj niemalejąco wylosowaną niepustą listę w tym wektorze (losuj indeks, aż do wylosowania pierwszej listy niepustej). Wyprowadź na konsolę zawartość listy przed i po uporządkowaniu.

Zadanie dla chętnych

5. Specjalsy

- Zdefiniuj strukturę generyczną (wzorzec!) Number do specjalnej reprezentacji dodatnich liczb całkowitych/rzeczywistych, zwanych dalej specjalsami. Oprócz pola Numb, umieść w strukturze pole wektorowe Dividers do przechowywania różnych (!) podzielników specjalsa oraz jednoargumentowy konstruktor z odpowiednią listą inicjalizacyjną.
- Zdefiniuj generyczną funkcję calculate_Dividers służącą do wyznaczania różnych podzielników specjalsa, a dokładnie - części całkowitej specjalsa. Wyznaczone podzielniki umieść w polu Dividers. Przyjmij, że 0 (zero) nie ma podzielników. Wynikiem działania funkcji ma być zmodyfikowany specjals.
- Zdefiniuj generyczny przeciążony operator "<<" służący do wyprowadzania specjalsa w postaci:

Number: 12.57

Dividers: 1 2 3 4 6 12

do wskazanego pliku. Uwaga: Wynik działania operatora ma być typu void.

- (d) Zdefiniuj generyczną funkcję `which_Number_greater` służącą do porównywania specjalsów. Z dwóch specjalsów ten jest większy, który ma większą liczbę podzielników. Wynikiem działania funkcji ma być wartość całkowita:
- 1 , w wypadku, gdy pierwszy specjals jest większy od drugiego,
 - 2, w przeciwnym wypadku.
- (e) Zdefiniuj generyczną funkcję `Add_Number_to_Tree`, dodającą wskazanego specjalsa do wskazanego drzewa BST specjalsów. Wynikiem działania funkcji ma być (inteligentny) wskaźnik do drzewa zmodyfikowanego w wyniku dodania specjalsa.
- (f) Zdefiniuj generyczny przeciążony operator `"<<"` służący do wyprowadzania zawartości wskazanego drzewa BST specjalsów, w porządku inorder, do wskazanego pliku. Po wyprowadzeniu każdego specjalsa, należy zrobić wiersz odstępu. Wynikiem działania operatora ma być referencja do przedmiotowego pliku.
- (g) W pliku `Specjalsy.txt` zadana jest pewna liczba dodatnich liczb rzeczywistych. Utwórz z nich drzewo BST specjalsów.
- Sprawdź przy użyciu wybranej funkcji algorytmicznej, czy wśród specjalsów jest taki, który ma 0 podzielników. Jeśli tak, to wyprowadź drzewo BST na standardowe wyjście.