

zaawansowane techniki C++ i STL**algorytmy i iteratory**

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1 (2 pkt)

Pojedyncza osoba jest opisana za pomocą imienia (`string`), nazwiska (`string`), wieku (`int`), wagi (`int`) mierzonej w kilogramach i wzrostu (`double`) mierzonego w metrach – zdefiniuj klasę `person` reprezentującą osobę. Dane o 12 osobach umieść w deku `deque<person>`. Wykonaj po kolejni następujące czynności:

- a. posortuj osoby według współczynnika BMI (funkcją porównującą niech będzie lambda); posortowaną grupę osób wypisz na standardowym wyjściu;
- b. osoby odchudziły się na obozie sportowym i straciły 10% swojej pierwotnej wagi (funkcją modyfikującą niech będzie lambda); zmodyfikowaną grupę osób wypisz na standardowym wyjściu;
- c. podziel osoby na dwie grupy: ciężkie z wagą powyżej 100[kg] i lekkie z wagą nie większą niż 100[kg] (użyj lambdy jako predykatu); wypisz grupę osób po podziale na standardowym wyjściu;
- d. wypisz osobę najstarszą i najmłodszą na standardowym wyjściu (nie korzystaj z sortowania danych).

Zadanie 2 (2 pkt)

Pojedynczy punkt jest opisany za pomocą współrzędnych, koloru w postaci RGB i nazwy – zdefiniuj klasę `point` reprezentującą punkt. Dane o 23 kolorowych punktach na płaszczyźnie umieść w liście `list<point>`. Wykonaj po kolejni następujące czynności:

- a. usuń z listy wszystkie punkty, których nazwy są dłuższe niż 5 znaków; wypisz pozostałe na liście punkty;
- b. policz ile jest punktów leżących w I, II, III i IV ćwiartce układu współrzędnych;
- c. posortuj punkty ze względu na ich jasność (luminancja koloru RGB jest określona wzorem $0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B$); wypisz punkty po posortowaniu;
- d. policz ile jest ciemnych punktów, dla których luminancja ma wartość poniżej 64; przenieś te punkty do innej listy i wypisz je.

Zadanie 3 (2 pkt)

Napisz funkcję, która zwróci najczęściej pojawiający się element w zbiorze danych oraz liczbę jego wystąpień. Dane to liczby całkowite umieszczone w wektorze `vector<int>`, które lubią się powtarzać. Jeśli więcej niż jeden element pojawia się pojawia się taką samą maksymalną liczbę razy, to funkcja powinna zwrócić wszystkie je wszystkie – na przykład dla danych {1, 1, 3, 5, 8, 9, 5, 8, 8, 5} powinny zostać zwrócone dwie pary {5, 3} oraz {8, 3} (pierwsza pozycja w parze to wartość a druga to liczba wystąpień).

Zadanie 4 (2 pkt)

Napisz program, który oblicza i wyświetla histogram zawierający częstość występowania liter alfabetu angielskiego (bez rozróżniania małych i dużych liter) w zadanym tekście. Tekst pobierz z pliku, którego nazwę przekąszes do programu poprzez argumenty wywołania. Częstość występowania danej litery jest zdefiniowana jako proporcja liczby wystąpień tej litery w stosunku do wszystkich liter w tekście (częstość musi być określona na podstawie liczby liter a nie wszystkich znaków w tekście). Wykorzystaj w swoim programie iterator strumieniowy.

Zadanie 5 (1 pkt)

Napisz funkcję, która wygeneruje wszystkie możliwe permutacje liter w słowie umieszczonym w obiekcie typu `string`. Zauważ, że takie same litery występujące w słowie są nieroóżnialne.

Zadanie 6 (1 pkt)

Pojedyncza osoba jest opisana za pomocą imienia (`string`), nazwiska (`string`), wieku (`int`), wagi (`int`) mierzonej w kilogramach i wzrostu (`double`) mierzonego w metrach – zdefiniuj klasę `person` reprezentującą osobę. Zdefiniuj trzy funkcje pracujące z osobą:

- pierwsza funkcja, która dla zadanej osoby zwróci tuplę z wartościami: imię i nazwisko osoby, jej wiek oraz wyliczony współczynnik BMI.
- druga funkcja, która dla zadanej osoby zwróci tuplę z wartościami: wiek osoby, jej waga oraz wzrost.
- trzecia funkcja, która dla zadanej osoby zwróci tuplę z wartościami: imię i nazwisko osoby oraz jej wiek.

Napisz program, w którym zostanie stworzony obiekt dla konkretnej osoby. Najpierw użyj pierwszej funkcji do obliczeń na tym obiekcie, odbierz wyniki za pomocą wiązania strukturalnego a potem wypisz odebrane wartości na standardowym wyjściu. Następnie użyj drugiej i trzeciej funkcji pracującej na tym obiekcie, ale wyniki odbierz w postaci tupli; odebrane wyniki wypisz na standardowym wyjściu iterując po tuplach.