Zadanie 1

Korzystając z zasobnikow sekwencyjnych oraz algorytmów ogólnych dostępnych w bibliotece STL

- wygenerować wektor 20 losowych liczb całkowitych z zakresu [30,79] (`generate`)
- usunąć z wektora wartości powtarzające się (`sort`,`unique`)
- wyświetlić zawartość wektora na standardowym wyjsciu przed i po operacji usuwania elementów

Zadanie 2

Korzystając z zasobnikow sekwencyjnych oraz algorytmów ogólnych dostępnych w bibliotece STL

- wygenerować wektor 200 losowych liczb ca lkowitych z zakresu [1,200]
- przenieść wszystkie wartości z zakresu [27,83] do listy wynikowej (skopiować i usunąć z orginalnej kolekcji `remove_if`, `remove_copy_if`)
- wyświetlić wektor zródłowy i listę docelową na standardowym wyjsciu

Zadanie 3

Korzystając z zasobnikow sekwencyjnych oraz algorytmów ogólnych dostępnych w bibliotece STL

- wygenerować 20 punktów w przestrzeni trójwymiarowej (generator nie jest ograniczony tylko do wartośći typów wbudowanych, może generować obiekty typów zdefiniowanych przez użytkownika), wartości współrzędnych punktow (liczby zmiennoprzecinkowe) powinny być generowane z przedziału [a,b),
- obliczyć objętości prostopadłoscianów jakie tworzą te punkty z początkiem układu wspłrzędnych (wygenerowany punkt). wartości objętości umieścić w oddzielnej strukturze danych (list, vector) (V = x*y*z) (`transform`),
- wyznaczyc średnią geometryczną objętość prostopadłościanów(`accumulate` z nagłówka `numeric`
- usunąc te punkty dla ktorych objętość tworzonego prostopadłościanu jest mniejsza od średniej.

```
vector<int> v1(10), v2;
copy (v1.begin(),v1.end(),back_inserter(v2));
```

Przykład pokazuje jak wstawiać elementy na końcu wektora (back_inserter działa nie tylko z copy)

Tym razem kontekstu testującego nie ma, sposób użycia można jednak łatwo wywnioskować z dokumentacji umieszczonej na stronie i poprzednich zadań. Dla ułatwienia w treści zadania podano jakie algorytmy należy użyć.