

# Programowanie II R

Zadania – seria 7.

STL: kontenery.

## Zadanie 1. lotto – Losowanie liczb.

Napisz program `lotto`, który losuje sześć różnych liczb całkowitych z przedziału od 1 do 49 włącznie, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej. Wykorzystaj `std::set<T>` oraz generator liczb losowych *Mersenne Twister 19937*, zaimplementowany jako typ `std::mt19937` w pliku nagłówkowym `random`.

### Przykładowe wykonanie

*Wywołanie:*

*Windows:* `lotto.exe`  
*macOS / Linux:* `./lotto`

*Wyjście:*

1 7 14 16 36 46

## Zadanie 2. intersection – Część wspólna zbiorów liczb.

Napisz funkcję

```
std::set<int> Intersection(const std::set<int> &a, const std::set<int> &b)
```

przyjmującą jako argumenty referencje `a` i `b` dwóch zbiorów liczb całkowitych oraz zwracającą zbiór stanowiący część wspólną tych zbiorów.

Korzystając z tej funkcji napisz program `intersection`, który wczytuje ze standardowego wejścia dwa ciągi liczb całkowitych oddzielonych znakami białymi, zapisane w osobnych liniach, i wypisuje na standardowe wyjście te liczby, które pojawiają się w obu ciągach, w kolejności rosnącej, każdą tylko raz.

### Przykładowe wykonanie

*Wywołanie:*

*Windows:* `intersection.exe`  
*macOS / Linux:* `./intersection`

*Wejście*

1 25 7 3 [Enter]  
3 4 -9 1 144 [Enter]

*Wyjście*

1 3

## Zadanie 3. words – Liczenie wyrazów.

Napisz program `words`, który wczytuje ze standardowego wejścia wyrazy aż do napotkania znaku końca pliku, a następnie wypisuje te wyrazy na standardowe wyjście w kolejności alfabetycznej, każdy tylko raz, podając przy każdym z nich ilość jego wystąpień. Wykorzystaj `std::map<Key, T>`.

### Przykładowe wykonanie

*Wywołanie:*

*Windows:* words.exe

*macOS / Linux:* ./words

*Wejście*

pies kot żółw kot kot pies żółw

*Wyjście*

kot 3

pies 2

żółw 2

### Zadanie 4. reverse – Odwracanie ciągu liczb.

Napisz program `reverse` wczytujący ze standardowego wejścia liczby całkowite aż do napotkania znaku końca pliku i wypisujący na standardowe wyjście te liczby w odwróconej kolejności. Wykorzystaj `std::stack<T>`.

### Przykładowe wykonanie

*Wywołanie:*

*Windows:* reverse.exe

*macOS / Linux:* ./reverse

*Wejście*

1 3 4 9 12

*Wyjście*

12 9 4 3 1

*Opracowanie: Bartłomiej Zglinicki.*