

## Wprowadzenie do STL (Standard Template Library)

## Zadanie 1

Korzystanie ze struktur standardowych typu `vector` i `list`.

- Utwórz tablicę służącą do przechowywania liczb całkowitych. Wykorzystaj standardową strukturę `vector` np.:

```
#include <vector>
using namespace std;
vector<int> tab;
```

- Wypełnij tablicę pewną ilością elementów wczytywanych z klawiatury lub generowanych losowo (funkcja `rand()`). Do wstawienia elementu do tablicy użyj metody `push_back` lub `insert` (sprawdź jak zmienia się rozmiar tablicy – funkcja `size`):

```
int d;
tab.push_back( d);      lub tab.insert( tab.end(), d);
cout << tab.size() << endl;
```

- Wydrukuj zawartość tablicy na ekran używając obiektu `iterator`:

```
vector<int>::iterator itr;
for ( itr=tab.begin(); itr!=tab.end(); ++itr)
cout << *itr << endl;
```

- Wydrukuj od końca zawartość tablicy na ekran używając obiektu `reverse_iterator`:

```
vector<int>::reverse_iterator ritr;
for ( itr=tab.rbegin(); itr!=tab.rend(); ++ritr)
cout << *ritr << endl;
```

- Usuń wszystkie elementy z tablicy `tab` przez wykorzystanie metody `clear()`. Sprawdź czy tablica jest rzeczywiście pusta przez sprawdzenie rozmiaru (`size()`) i metody `empty()` – zwraca wartość `bool` w zależności od tego czy tablica jest pusta czy też nie.
- Zmodyfikuj program tak aby zamiast kolekcji typu `vector` korzystać z kolekcji typu `list` (pamiętaj o `include`).

## Zadanie 2

Korzystanie ze standardowych algorytmów.

- Posortuj rosnąco i wydrukuj zawartość tablicy `tab`. Skorzystaj z funkcji `sort` (pamiętaj o `#include <algorithm>`):

```
sort( tab.begin(), tab.end() );
Posortuj malejąco i wydrukuj zawartość tablicy tab:
sort( tab.begin(), tab.end(), greater<int>() );
```

- Czy efekt z punktu poprzedniego można osiągnąć inaczej? Czy funkcja `sort` będzie działała dla kolekcji typu `list` (dlaczego)?
- Znajdź największy i najmniejszy element w tablicy `tab` (funkcje `max_element` i `min_element`).
- Czy możesz zastosować algorytm dla tablicy typowej dla języka C? Sprawdź działanie w programie..
- Utwórz kolekcję `vector` składającą się z obiektów typu `Wektor2D`. Współrzędne wektorów 2D generuj losowo.
- Wydrukuj wszystkie elementy kolekcji `vector` posortowane rosnąco.
- Poeksperymentuj z innym algorytmami STL (opis znajdziesz w helpie pod hasłem „Standard C++ Library, algorithms”)