Звіт

По лабораторній роботі №5

Виконав:

Студент ФІТ 2-11

Забродський Олексій Русланович

**Завдання 1**

1.Створити об'єкт-контейнер відповідно до варіанта завдання і заповнити його даними, тип яких визначається варіантом завдання.

2.Переглянути контейнер.

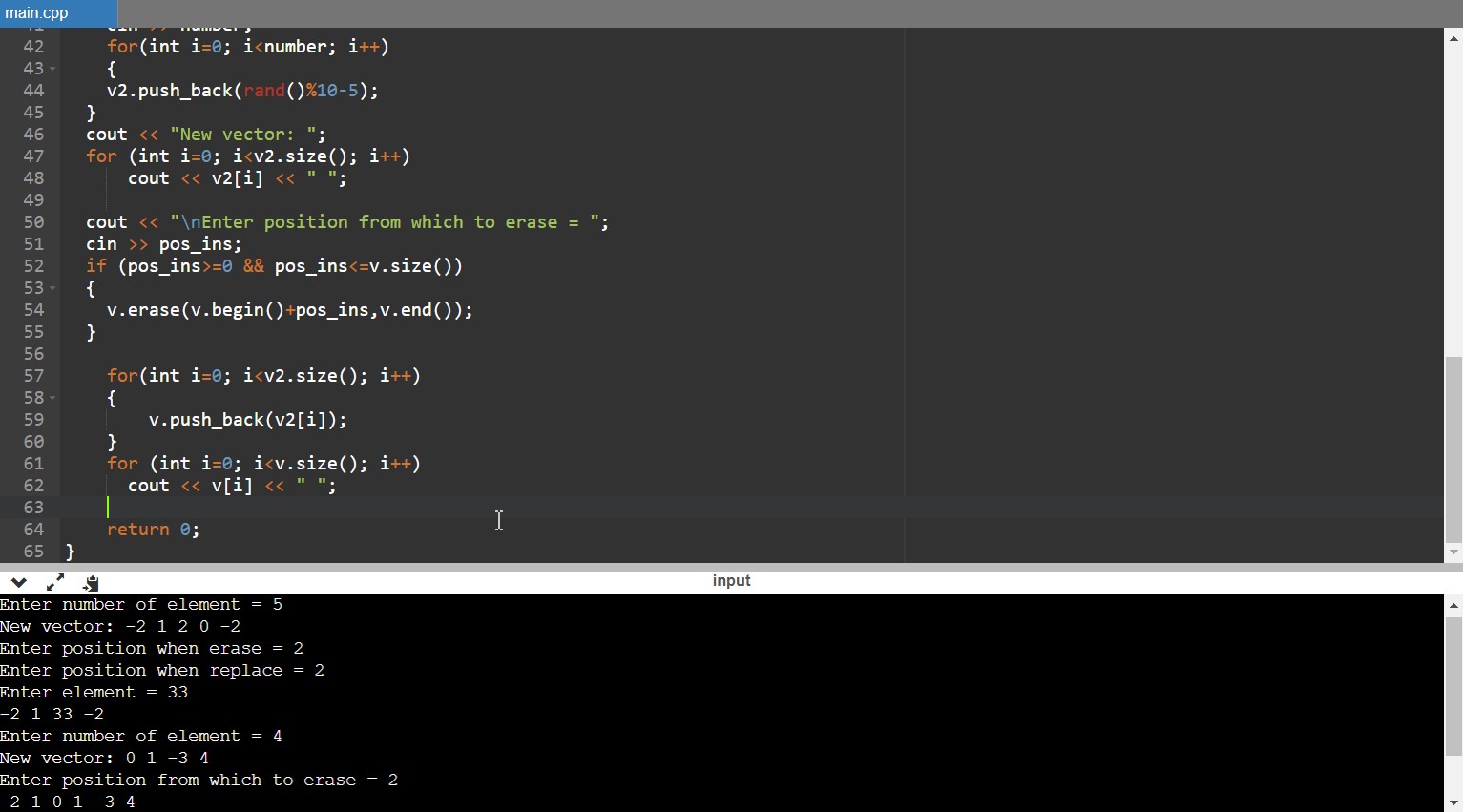
3.Змінити контейнер, видаливши з нього одні елементи і замінивши інші.

4.Переглянути контейнер.

5.Створити другий контейнер цього ж класу і заповнити його даними того ж типу, що і перший контейнер.

6.Змінити перший контейнер, видаливши з нього n елементів після заданого і додавши потім у нього всі елементи з другого контейнера.

Варіант 8. типу float без використання ітератора



**Лістинг програми:**

#include <iostream>

#include <vector>

using namespace std;

int main()

{

vector <float> v;

int number, pos\_ins;

float data;

cout << "Enter number of element = ";

cin >> number;

for(int i=0; i<number; i++)

{

v.push\_back(rand()%10-5);

}

cout << "New vector: ";

for (int i=0; i<v.size(); i++)

cout << v[i] << " ";

cout << "\nEnter position when erase = ";

cin >> pos\_ins;

if (pos\_ins>=0 && pos\_ins<=v.size())

{

v.erase(v.begin()+pos\_ins);

}

cout << "Enter position when replace = ";

cin >> pos\_ins;

cout << "Enter element = ";

cin >> data;

v[pos\_ins] = data;

for (int i=0; i<v.size(); i++)

cout << v[i] << " ";

vector<float> v2;

cout << "\nEnter number of element = ";

cin >> number;

for(int i=0; i<number; i++)

{

v2.push\_back(rand()%10-5);

}

cout << "New vector: ";

for (int i=0; i<v2.size(); i++)

cout << v2[i] << " ";

cout << "\nEnter position from which to erase = ";

cin >> pos\_ins;

if (pos\_ins>=0 && pos\_ins<=v.size())

{

v.erase(v.begin()+pos\_ins,v.end());

}

for(int i=0; i<v2.size(); i++)

{

v.push\_back(v2[i]);

}

for (int i=0; i<v.size(); i++)

cout << v[i] << " ";

return 0;

}

**Вхідні дані програми:**

Enter number of element = 5

Enter position when erase = 2

Enter position when replace = 2

Enter element = 33

Enter number of element = 4

Enter position from which to erase = 2

**Вихідні дані:**

New vector: -2 1 2 0 -2

-2 1 33 -2

New vector: 0 1 -3 4

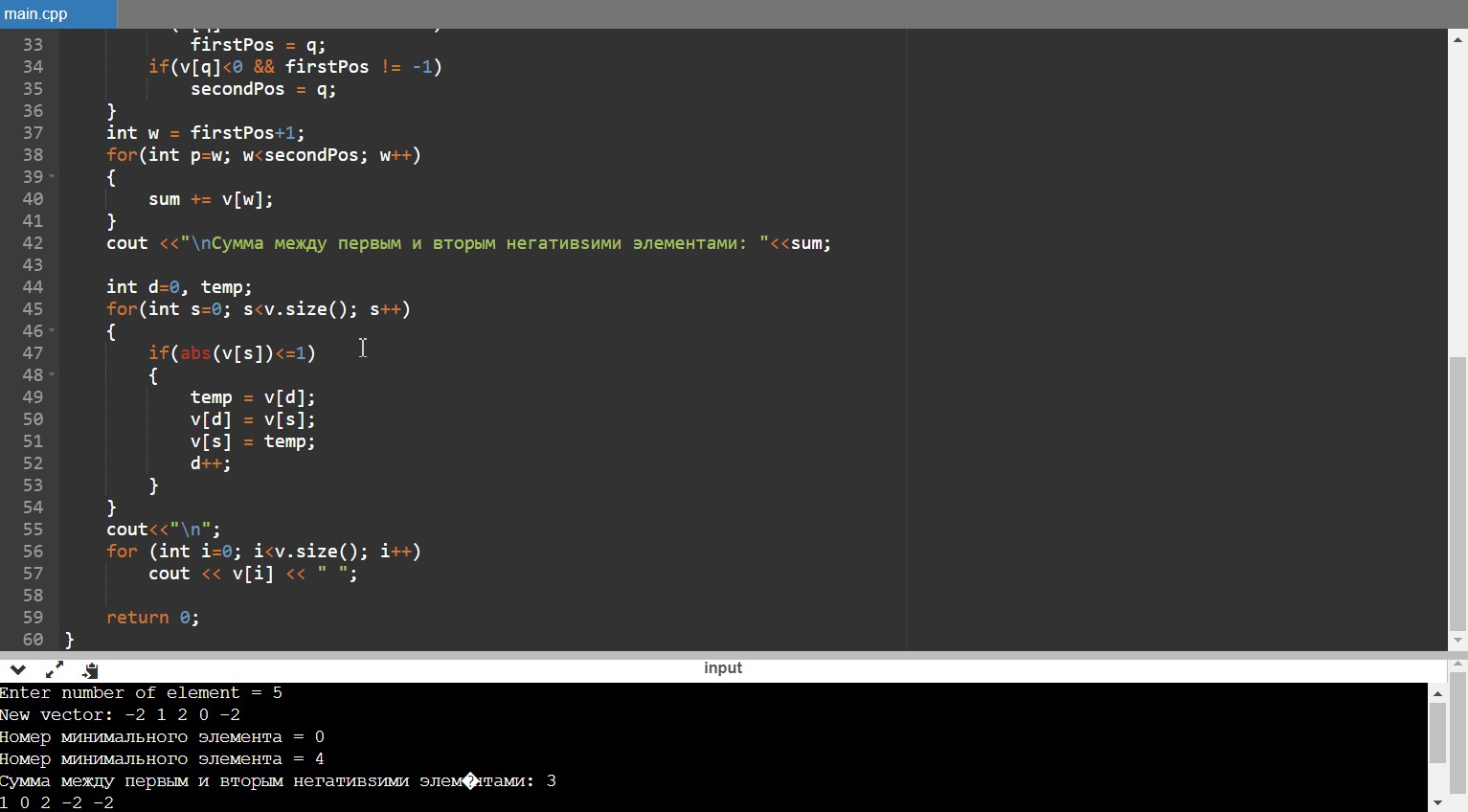
-2 1 0 1 -3 4

**Завдання 2**

##### Варіант 8. В одномірному масиві, що складається з n елементів, обчислити:

* номер мінімального елемента масиву;
* суму елементів масиву, розташованих між першим і другим негативними елементами;

перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, модуль яких не перевищує 1, а потім — всі інші.



**Лістинг програми:**

#include <iostream>

#include <vector>

#include <cmath>

#include <algorithm>

using namespace std;

int main()

{

vector <int> v;

int number, sum=0;

cout << "Enter number of element = ";

cin >> number;

for(int i=0; i<number; i++)

{

v.push\_back(rand()%10-5);

}

cout << "New vector: ";

for (int i=0; i<v.size(); i++)

cout << v[i] << " ";

std::vector<int>::iterator min;

min = min\_element(v.begin(),v.end());

for(int s=0; s<v.size(); s++)

if(v[s]==\*min)

cout <<"\nНомер минимального элемента = "<< s;

int firstPos = -1, secondPos = -1;

for(int q=0; q<v.size(); q++)

{

if(v[q]<0 && firstPos == -1)

firstPos = q;

if(v[q]<0 && firstPos != -1)

secondPos = q;

}

int w = firstPos+1;

for(int p=w; w<secondPos; w++)

{

sum += v[w];

}

cout <<"\nСумма между первым и вторым негативsими элементами: "<<sum;

int d=0, temp;

for(int s=0; s<v.size(); s++)

{

if(abs(v[s])<=1)

{

temp = v[d];

v[d] = v[s];

v[s] = temp;

d++;

}

}

cout<<"\n";

for (int i=0; i<v.size(); i++)

cout << v[i] << " ";

return 0;

}

**Вхідні дані програми:**

Enter number of element = 5

**Вихідні дані:**

New vector: -2 1 2 0 -2

Номер минимального элемента = 0

Номер минимального элемента = 4

Сумма между первым и вторым негативsими элемэнтами: 3

1 0 2 -2 -2

**Висновок**

Виконуючи лабораторну роботу я набув практичних навичок при роботі з контейнерами стандартної бібліотеки шаблонів STL. Ознайомився з особливостями розробки класів для роботи з динамічними масивами довільної розмірності з використанням контейнерів vector та algorithm стандартної бібліотеки шаблонів STL.