**Gr no 21810819**

**Assignment No:7**

**Problem statement:**

Using standard template library (STL) list container implement following member functions of list class: empty, insert, reverse, sort, unique, using iterator.

**Aim of the Assignment:**

To implement various concept in standard template library list container to perform Various member function of list.

**Description:**

A Standard Template Library list container is defined. Some member functions are used like insert, reverse, sort, unique of the list class. Iterators are used to point at the memory addresses of STL containers.Program is made menu driven to perform the task as per user’s choice.

**OOP concept used:**

List container:- Lists are sequence containers that allow non-contiguous memory allocation. As compared to vector, list has slow traversal, but once a position has been found, insertion and deletion are quick. Normally, when we say a List, we talk about doubly linked list. For implementing a singly linked list, we use forward list.

Functions in STL List:

empty () – Returns whether the list is empty(1) or not(0).

reverse():reverse the list.

merge(): merge the two list.

Unique: removes the duplicate elements in list.

Iterator:- Iterators are used to point at the memory addresses of [STL](http://quiz.geeksforgeeks.org/the-c-standard-template-library-stl/) containers. They are primarily used in sequence of numbers, characters etc. They reduce the complexity and execution time of program.

**Code:**

#include<iostream>

using namespace std;

#include <list>

#include <iterator>

using namespace std;

//function for printing the elements in a list

int showlist(list <int> g)

{

list <int> :: iterator it;

for(it = g.begin(); it != g.end(); ++it)

cout << '\t' << \*it;

cout << '\n';

}

int main()

{

list <int> l1;

cout<<"List status:"<<l1.empty();

for (int i = 0; i < 10; ++i)

{

l1.push\_back(i \* 2);

}

cout << "\nList 1 (l1) is : ";

showlist(l1);

cout << "\nlist1.front() : " << l1.front();

cout << "\nlist1.back() : " << l1.back();

cout << "\nlist1.pop\_front() : ";

l1.pop\_front();

showlist(l1);

cout << "\nlist1.reverse() : ";

l1.reverse();

showlist(l1);

cout<<"\nSorting():";

l1.sort();

showlist(l1);

cout<<"List status:"<<l1.empty();

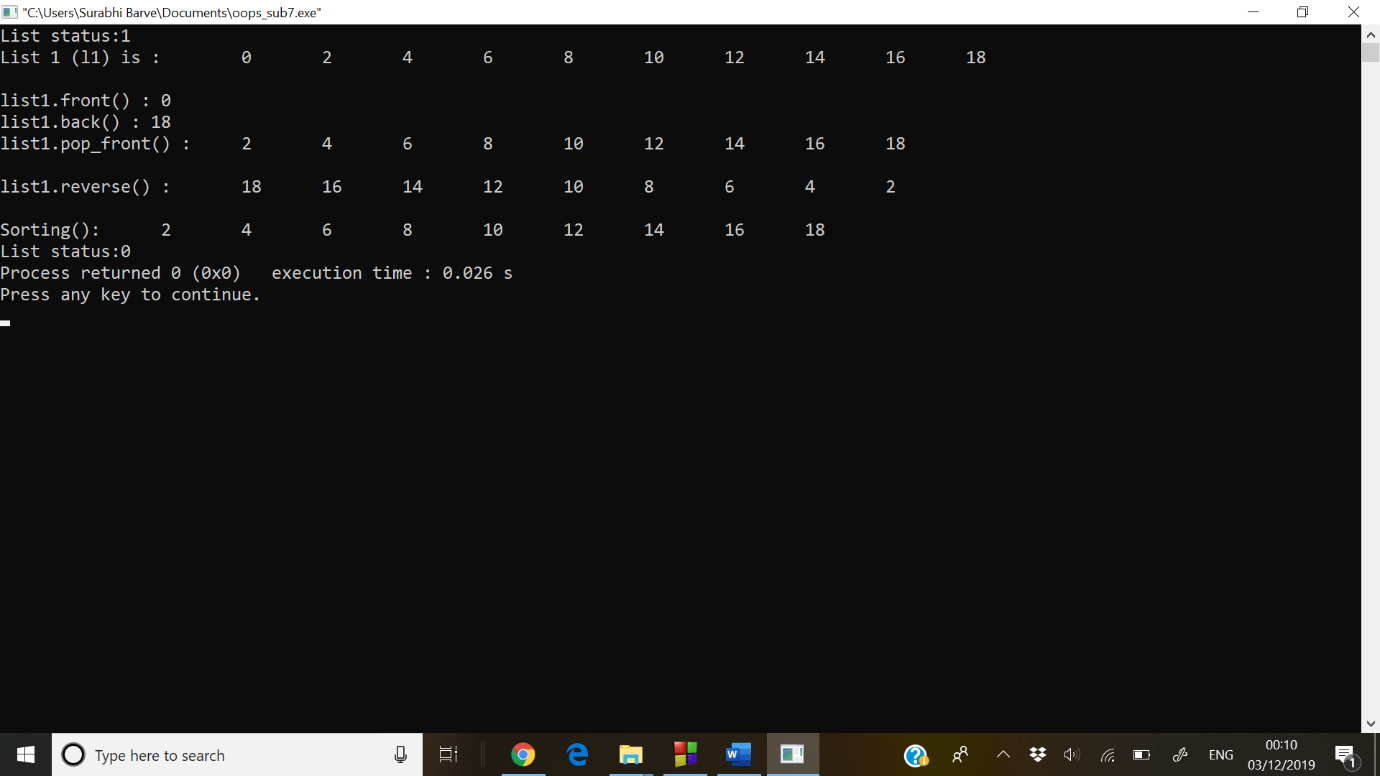
return 0;

}

**Conclusion:**

The various operations on STL list are implemented successfully.

**Screenshot:**

****