**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 Assignment:** To implement the STL concept in C++ using list and perform various operations on list.

**Description:** A list is defined. Various different functions of list are used such as empty(), pushback(),sort(), unique() and merge(). A menu driven program is used displaying various operations on list.

**OOP Concept used:**

1. List:- 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.

[push\_back(g)](https://www.geeksforgeeks.org/list-push_back-function-in-c-stl/) – Adds a new element ‘g’ at the end of the list.

[empty()](https://www.geeksforgeeks.org/list-empty-function-in-c-stl/) – Returns whether the list is empty(1) or not(0).

[sort()](https://www.geeksforgeeks.org/stdlistsort-c-stl/) – Sorts the list in increasing order.

[reverse()](https://www.geeksforgeeks.org/list-reverse-function-in-c-stl/) – Reverses the list.

[list merge() function in C++ STL](https://www.geeksforgeeks.org/list-merge-function-in-c-stl/)– Merges two sorted lists into one

2. 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

**Conclusion:** The various operations on list are implemented successfully.

**Program:**

#include<iostream>

#include<list>

#include<stdlib.h>

using namespace std;

void display(list<int> &l)

{

list<int>::iterator itr;

for(itr=l.begin(); itr!=l.end(); ++itr)

{

cout<<\*itr<<" ";

}

}

int main()

{

int x,pos,no,elem;

int value;

list<int>l1;

list<int>l2;

for(int i=0;i<5;i++)

{

cout<<"ENTER THE VALUE FOR 1st LIST"<<'\n';

cin>>value;

l1.push\_back(value);

}

system("CLS");

for(int i=0;i<1000;i++)

{

cout<<"WHAT FUNCTION DO YOU WANNT TO PERFORM"<<'\n';

cout<<"1.DISPLAY 1st LIST"<<'\n';

cout<<"2.SORT 1st LIST"<<'\n';

cout<<"3.MERGE THE LISTS"<<'\n';

cout<<"4.CHECK IF THE LIST ARE EMPTY"<<'\n';

cout<<"5.INSERT ELEMENTS IN LIST"<<'\n';

cout<<"6.REVERSE THE LIST"<<'\n';

cout<<"7.REMOVE THE REPEATATING ELEMENTS IN LIST"<<'\n';

cout<<"8.EXIT"<<'\n';

cin>>x;

switch(x)

{

case 1:

display(l1);

cout<<'\n';

break;

case 2:

l1.sort();

display(l1);

cout<<'\n';

break;

case 3:

for(int i=0;i<5;i++)

{

cout<<"ENTER THE VALUE FOR 2nd LIST"<<'\n';

cin>>value;

l2.push\_back(value);

}

l1.merge(l2);

display(l1);

break;

case 4:

if(l1.empty()==true)

{

cout<<"LIST IS EMPTY"<<'\n';

}

else

{

cout<<"LIST IS NOT EMPTY"<<'\n';

}

break;

case 5:

{

list<int>::iterator it1=l1.begin();

cout<<"ENTER THE POSITION AT WHICH YOU WANT TO INSERT A NO."<<'\n';

cin>>pos;

advance(it1,pos-1);

cout<<"ENTER THE ELEMENTS"<<'\n';

cin>>elem;

l1.insert(it1,elem);

display(l1);

}

break;

case 6:

l1.reverse();

display(l1);

break;

case 7:

l1.unique();

display(l1);

break;

case 8:

return 0;

}

}

}









