ASSIGNMENT 7

Aim:-

Using standard template library (STL) list container implement following member functions of list class:

* Empty
* Insert
* Reverse
* Sort
* Unique
* Iterator

Objective:

To understand different type of operation can perform on list. Learn different type member function used on list.

Theory:

**Empty()**:

The **list::empty()** is a built-in function in C++ STL is used to check whether a particular list container is empty or not. This function does not modifies the list, it simply checks whether a list is empty or not, i.e. the size of list is zero or not.

**Syntax:**

list\_name.empty()

**Insert():**

The **list::insert()** is used to insert the elements at any position of list. This function takes 3 elements, position, number of elements to insert and value to insert. If not mentioned, number of elements is default set to 1.

**Syntax:**

insert(pos\_iter, ele\_num, ele)

**Reverse():**

reverse() is a predefined function in header file algorithm. It is defined as a template in the above mentioned header file. It reverses the order of the elements in the range [first, last) of any container.

**void reverse (BidirectionalIterator first, BidirectionalIterator last)**

**Sort():**

We have discussed [qsort() in C.](https://www.geeksforgeeks.org/comparator-function-of-qsort-in-c/) C++ STL provides a similar function sort that sorts a vector or array (items with random access). Below is a simple program to show working of sort().

**Unique:**

**std::unique** is used to remove duplicates of any element present consecutively in a range[first, last). It performs this task for all the sub-groups present in the range having the same element present consecutively.

**template**

**ForwardIterator unique (ForwardIterator first, ForwardIterator last);**

**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.

**Operations of iterators** :-

1. **begin()** :- This function is used to return the **beginning position** of the container.
2. **end()** :- This function is used to return the**after end position** of the container.

Sourcecode:

#include<iostream>

#include<list>

using namespace std;

int display(list<int> &l)

{

list<int>::iterator itr;

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

{

cout<<\*itr;

cout<<" ";

}

return 0;

}

int main()

{

int value;

list<int> l1(3);

list<int> l2(5);

cout<<"\n\nAdd at the back\n";

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

{

cin>>value;

l1.push\_back(value);

l2.push\_back(value+5);

}

display(l1);

cout<<" ";

display(l2);

cout<<" ";

cout<<"\n\nSort list1 and list2\n";

l1.sort();

l2.sort();

display(l1);

cout<<" ";

display(l2);

cout<<" ";

cout<<"\n\nMerge two lists\n";

l1.merge(l2);

display(l1);

cout<<" ";

cout<<"\n\nAdd at the front\n";

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

{

cin>>value;

l1.push\_front(value);

l2.push\_front(value+5);

}

display(l1);

cout<<" ";

display(l2);

cout<<" ";

return 0;

}

Output

/\*output:

Add at the back

6

5

5

45

5

0 0 0 6 5 5 45 5 0 0 0 0 0 11 10 10 50 10

Sort list1 and list2

0 0 0 5 5 5 6 45 0 0 0 0 0 10 10 10 11 50

Merge two lists

0 0 0 0 0 0 0 0 5 5 5 6 10 10 10 11 45 50

Add at the front

2

5

5

6

3

3 6 5 5 2 0 0 0 0 0 0 0 0 5 5 5 6 10 10 10 11 45 50 8 11 10 10 7

--------------------------------

Process exited after 43.41 seconds with return value 0

Press any key to continue . .\*/

**Conclusion:**

Class list have different type of function. We can check list empty or not.

Reverse, sort and insert element any position.