List

List is a contiguous container while vector is a non-contiguous container i.e list stores the elements on a contiguous memory and vector stores on a non-contiguous memory.

Insertion and deletion in the middle of the vector is very costly as it takes lot of time in shifting all the elements. Linklist overcome this problem and it is implemented using list container.

List supports a bidirectional and provides an efficient way for insertion and deletion operations.

Traversal is slow in list as list elements are accessed sequentially while vector supports a random access.

Template for list

```
#include<iostream>

#include<list>

using namespace std;

int main()

{

  list<int> l;

}
```

It creates an empty list of integer type values.

List can also be initalised with the parameters.

```
#include<iostream>

#include<list>

using namespace std;

int main()

{
```

```
   list<int> l{1,2,3,4};
}
```

List can be initialised in two ways.

```
list<int>  new_list{1,2,3,4};
```

or

```
list<int> new_list = {1,2,3,4};
```

C++ List Functions

Following are the member functions of the list:

Method	Description
insert()	It inserts the new element before the position pointed by the iterator.
push_back()	It adds a new element at the end of the vector.
push_front()	It adds a new element to the front.
pop_back()	It deletes the last element.
pop_front()	It deletes the first element.
empty()	It checks whether the list is empty or not.
size()	It finds the number of elements present in the list.
max_size()	It finds the maximum size of the list.
front()	It returns the first element of the list.
back()	It returns the last element of the list.
swap()	It swaps two list when the type of both the list are same.
reverse()	It reverses the elements of the list.
sort()	It sorts the elements of the list in an increasing order.
merge()	It merges the two sorted list.
splice()	It inserts a new list into the invoking list.
unique()	It removes all the duplicate elements from the list.
resize()	It changes the size of the list container.
assign()	It assigns a new element to the list container.
emplace()	It inserts a new element at a specified position.
emplace_back()	It inserts a new element at the end of the vector.
emplace_front()	It inserts a new element at the beginning of the list.

Example: C++ program to demonstrate the implementation of List

```
#include <iostream>
#include <iterator>
#include <list>
using namespace std;

// function for printing the elements in a list
void showlist(list<int> g)
{
	list<int>::iterator it;
	for (it = g.begin(); it != g.end(); ++it)
		cout << '\t' << *it;
	cout << '\n';
}

// Driver Code
int main()
{

	list<int> gqlist1, gqlist2;

	for (int i = 0; i < 10; ++i) {
		gqlist1.push_back(i * 2);
		gqlist2.push_front(i * 3);
	}
	cout << "\nList 1 (gqlist1) is : ";
	showlist(gqlist1);

	cout << "\nList 2 (gqlist2) is : ";
	showlist(gqlist2);

	cout << "\ngqlist1.front() : " << gqlist1.front();
	cout << "\ngqlist1.back() : " << gqlist1.back();

	cout << "\ngqlist1.pop_front() : ";
	gqlist1.pop_front();
	showlist(gqlist1);
```

```
    cout << "\ngqlist2.pop_back() : ";
    gqlist2.pop_back();
    showlist(gqlist2);

    cout << "\ngqlist1.reverse() : ";
    gqlist1.reverse();
    showlist(gqlist1);

    cout << "\ngqlist2.sort(): ";
    gqlist2.sort();
    showlist(gqlist2);

    return 0;
}
```