Assignment - 38 **A Job Ready Bootcamp in C++, DSA and IOT**

list

1. List functions in C++ STL (Standard Template Library)

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

int main()
{
    //declare and initialize lists
    list<int> list1 {15, 8, 100, 25, 58};
    list<int> list2;

    //check list1 is empty or not
    if(list1.empty())
        cout<<"list1 is an empty list\n";
    else
        cout<<"list1 is not an empty list\n";

    //check list2 is empty or not
    if(list2.empty())
        cout<<"list2 is an empty list\n";
    else
        cout<<"list2 is not an empty list\n";

    cout << "Size of list: " << list1.size();

    list1.sort();
    cout << "\nSort" << endl;
    for(auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";

    list1.reverse();
    cout << "\nReverse" << endl;
    for(auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";

    list1.remove(100);
    cout << "\nRemove 100" << endl;
    for(auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";

    cout << "\nFront: " << list1.front() << endl;
    cout << "Back: " << list1.back() << endl;

    list1.push_front(99);
    list1.push_back(39);
    for(auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";

    cout << endl;
    list1.pop_front();
    list1.pop_back();
    for(auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";

    cout << "\nList 2: ";
    list2 = {10, 20, 30, 40, 50, 60};
    for(auto it = list2.begin(); it != list2.end(); it++)
        cout << *it << " ";

    cout << "\nAfter mearge: ";
```

```
    list1.merge(list2);
    for(auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";
    return 0;
}
```

```
Output:
list1 is not an empty list
list2 is an empty list
Size of list: 5
Sort
8 15 25 58 100
Reverse
100 58 25 15 8
Remove 100
58 25 15 8
Front: 58
Back: 8
99 58 25 15 8 39
58 25 15 8
List 2: 10 20 30 40 50 60
After mearge: 10 20 30 40 50 58 25 15 8 60
```

S.No.	Function	Description	Syntax (consider 'list' is the name of any integer list)
1.	empty()	Checks whether the given list is empty or not. It returns 1 (true) if the list is empty else it returns 0 (false).	list.empty();
2.	size()	Returns size of the list	list.size();
3.	sort()	Sorts the list in ascending order	list.sort();
4.	reverse()	Reverses the list (elements of the list)	list.reverse();
5.	remove()	Removes all occurrences of given elements from the list.	list.remove(element);
6.	remove_if()	Remove a set of the elements based on the test condition (if the test condition is true for the element, element will be removed and it is applicable on all the occurrences of the element which satisfy the test condition).	list.remove_if(test_condition);

7.	front()	Returns the first element of the list	list.front();
8.	back()	Returns the last element of the list	list.back();
9.	push_front()	Inserts the element at the front (beginning) to the list	list.push_front(element);
10.	push_back()	Insert the element at the back (end) to the list	list.push_back(element);
11.	pop_front()	Removes the element from the front (beginning) of the list	list.pop_front();
12.	pop_back()	Removes the element from the back (end) of the list	list.pop_back();
13.	insert()	Inserts the element at specified index/position	list.insert(iterator_position, element); OR list.insert(iterator_positon, number_of_elements, element);
14.	begin() and end()	Return the iterator pointing first and last element of the list	list.begin(); and list.end();
15.	rbegin() and rend()	Return the iterator pointing first and last element of the list (in reverse order) i.e. first element will be considered as last and last will be consider as first	list.rbegin(); and list.rend();
16.	assign()	Assigns the new set of elements or replaces the current with the new set of elements	list.assign(n, element) //will assign 'element', 'n' times to the list
17.	merge()	It merges two lists.	list1.merge(list2);
18.	unique()	It removes consecutive elements from the list.	list.unique();
19.	erase()	It removes the	list.erase(iterator_posi

		specified index or index from the given range (index1, index2); this function can be used by defining the positions using an iterator.	tion); OR list.erase(iterator_position1, iterator_position2);

2. Assign the elements to the list (different methods) - Example of list::assign() | C++ STL

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

int main()
{
    // first method to assign element to the list
    list<int> l1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    list<int>::iterator it;
    for (it = l1.begin(); it != l1.end(); it++)
        cout << *it << " ";
    cout << endl;

    // Second method to assign element to the list
    list<int> l2;
    int n;
    list<int>::iterator it2;
    cout << "Enter 10 elements: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n;
        l2.push_back(n);
    }
    for (it2 = l2.begin(); it2 != l2.end(); it2++)
        cout << *it2 << " ";
    cout << endl;

    // Third method to assign element to the list
    list<int> l3;
    int n1;
    list<int>::iterator it3;
    cout << "Enter 10 elements: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n1;
        l3.push_front(n1);
    }
    l3.reverse();
    for (it3 = l3.begin(); it3 != l3.end(); it3++)
        cout << *it3 << " ";
    cout << endl;
    return 0;
}
```

```
Output:
1 2 3 4 5 6 7 8 9 10
Enter 10 elements: 10 20 30 40 50 60 70 80 90 100
10 20 30 40 50 60 70 80 90 100
Enter 10 elements: 2 4 6 8 10 12 14 16 18 20
2 4 6 8 10 12 14 16 18 20
```

3. Iterate a list C++ STL

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

int main()
{
    list<int> l1;
    int n1;
    //assign through push_back
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n1;
        l1.push_back(n1);
    }
    list<int>::iterator it1;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;

    list<int> l2;
    int n2;
    //assign through push_front
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n2;
        l2.push_front(n2);
    }
    list<int>::iterator it2;
    for (it2 = l2.begin(); it2 != l2.end(); it2++)
        cout << *it2 << " ";
    cout << endl;

    return 0;
}
```

```
================================================================================
Output:
Enter 10 numbers: 1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Enter 10 numbers: 10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10
```

4. Iterate a list in reverse order C++ STL

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

int main()
{
    list<int> l1;
    int n1;
    //assign through push_back
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n1;
        l1.push_back(n1);
    }
    list<int>::iterator it1;
    l1.reverse();
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;
```

```
    list<int> l2;
    int n2;
    //assign through push_front
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n2;
        l2.push_front(n2);
    }
    list<int>::iterator it2;
    // l2.reverse();
    for (it2 = l2.begin(); it2 != l2.end(); it2++)
        cout << *it2 << " ";
    cout << endl;

    return 0;
}
```

```
Output:
Enter 10 numbers: 1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
Enter 10 numbers: 10 20 30 40 50 60 70 80 90 100
100 90 80 70 60 50 40 30 20 10
```

5. Input and add elements to a list C++ STL

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

int main()
{
    //declaring a list
    list<string> l1;
    //declaring iterator to the list
    list<string>::iterator it;

    //declaring string object
    string str;

    //input strings
    while(true)
    {
        cout<<"Enter string (\"ESC or ecs\" to quit): ";
        getline(cin, str);
        if(str=="ESC" || str == "esc")
            break;

        //adding string to the list
        l1.push_back(str);
    }

    //printing list elements
    cout<<"List elements are"<<endl;
    for (it = l1.begin(); it != l1.end(); it++)
        cout<< *it<<endl;

    return 0;
}
```

```
Output:
Enter string ("ESC or ecs" to quit): Akhtar
Enter string ("ESC or ecs" to quit): Mukesh
```

```
Enter string ("ESC or ecs" to quit): Gautam
Enter string ("ESC or ecs" to quit): Tarun
Enter string ("ESC or ecs" to quit): esc
List elements are
Akhtar
Mukesh
Gautam
Tarun
```

6. Get the first and last element of the list C++ STL

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

int main()
{
    list<int> l1;
    int n1;
    //assign through push_back
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n1;
        l1.push_back(n1);
    }

    list<int>::iterator it1;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;

    cout << "First Element : " << l1.front() << endl;
    cout << "Last Element : " << l1.back() << endl;
    return 0;
}
```

```
================================================================================
Output:
Enter 10 numbers: 10 20 30 40 50 60 70 80 90 100
10 20 30 40 50 60 70 80 90 100
First Element : 10
Last Element : 100
```

7. Insert the element at beginning and end of the list | C++ STL

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

int main()
{
    list<int> l1;
    int n1;
    //assign through push_back
```

```
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n1;
        l1.push_back(n1);
    }

    l1.push_front(10);
    l1.push_back(50);

    list<int>::iterator it1;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;

    return 0;
}
```

```
Output:
Enter 10 numbers: 8 16 24 32 40 48 56 64 72 80
10 8 16 24 32 40 48 56 64 72 80 50
```

8. Remove all occurrences of an element and remove set of some specific from the list C++ STL

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

int main()
{
    // declaring a list
    list<int> l1 = {11, 22, 33, 44, 55, 11, 22};
    // declaring iterator to the list
    list<int>::iterator it1;

    // printing list elements
    cout << "List elements are" << endl;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";

    // remove 11 from the List
    l1.remove(11);
    cout << "\nList elements after removing 11" << endl;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";

    // remove all ODD numbers
    l1.remove_if([](int n){ return (n % 2 != 0); });
    cout << "\nList elements after removing all ODD numbers" << endl;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";

    return 0;
}
```

```
Output:
List elements are
11 22 33 44 55 11 22
List elements after removing 11
22 33 44 55 22
List elements after removing all ODD numbers
22 44 22
```

9. Remove all consecutive duplicate elements from the list | C++ STL

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

int main()
{
    list<int> l1;
    int n1;
    //assign through push_back
    cout << "Enter 10 numbers: ";
    for (int i = 0; i < 10; i++)
    {
        cin >> n1;
        l1.push_back(n1);
    }

    list<int>::iterator it1;

    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;

    l1.unique();

    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;
    return 0;
}
```

```
Ouput:
Enter 10 numbers: 1 1 2 3 5 5 6 7 9 10
1 1 2 3 5 5 6 7 9 10
1 2 3 5 6 7 9 10
```

10. Merge two lists C++ STL

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

int main()
{
    list<int> list1 = {10, 20, 30, 40, 50};
    list<int> list2 = {1, 2, 3, 4, 5};

    // merge operation
    list1.merge(list2);

    cout << "List: ";

    for (auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";

    return 0;
}
```

```
Output:
List: 1 2 3 4 5 10 20 30 40 50
```

11. Creating a list by assigning the all elements of another list C++ STL

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

```
void printList(list<int> L)
{
    list <int> :: iterator it;
    for(auto it = L.begin(); it != L.end(); it++)
        cout << *it << " ";
    cout << endl;
}

int main()
{
    list <int>l1 = {1,2,3,4,5,6,7,8,9,10};
    list <int>l2;
    printList(l1);
    l2.assign(l1.begin(), l1.end());
    printList(l2);

    return 0;
}
```

```
Output:
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
```

12. Assign a list with array elements C++ STL

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

void printList(list<int> L)
{
    list<int>::iterator it;
    for (auto it = L.begin(); it != L.end(); it++)
        cout << *it << " ";
    cout << endl;
}

int main()
{
    array<int, 10> a1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    list<int> l2;

    l2.assign(a1.begin(), a1.end());
    printList(l2);

    return 0;
}
```

```
Output:
1 2 3 4 5 6 7 8 9 10
```

13. Push characters in a list and print them separated by space in C++ STL

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

void printList(list<char> L)
{
    list<int>::iterator it;
    for (auto it = L.begin(); it != L.end(); it++)
        cout << *it << " ";
```

```
    cout << endl;
}

int main()
{
    list<char> l2 = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'};

    printList(l2);

    return 0;
}
```

```
Output:
A B C D E F G H I
```

14. Access elements of a characters list using const_iterator in C++ STL

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

void printList(list<char> L)
{
    list<int>::iterator it;
    for (auto it = L.begin(); it != L.end(); it++)
        cout << *it << " ";
    cout << endl;
}

int main()
{
    list<char> l;
    char ch;
    cout << "Enter character list seperate by space" << endl;
    for (char i = 'A'; i <= 'Z'; i++)
    {
        cin >> ch;
        l.push_back(ch);
    }

    printList(l);

    return 0;
}
```

```
Output:
Enter character list seperate by space
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```