Assignment - 39 **A Job Ready Bootcamp in C++, DSA and IOT**

forward_list

1. Write a c++ code, to demonstrate the forward list.

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

int main()
{
    forward_list<int> l1;   //  create fordward_list
    l1.assign({1, 2, 3, 4, 5, 6});  //  assign in list
    forward_list<int>::iterator it;
    for (auto it = l1.begin(); it != l1.end(); it++)
        cout << *it << " ";

    return 0;
}
================================================================================
Output:
1 2 3 4 5 6
```

2. Write a c++ code, in which create a forward list and assign values to it at the time of initialization and print it on the console screen.

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

int main()
{
    forward_list<int> l1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // initilazation
forward_list
    forward_list<int>::iterator it;
    for (auto it = l1.begin(); it != l1.end(); it++)
        cout << *it << " ";

    return 0;
}
================================================================================
Output:
1 2 3 4 5 6 7 8 9 10
```

3. Create a forward list insert elements from 1 to 10 and find the sum of elements.

```
#include <iostream>
#include <forward_list>
#include <iterator>
#include <numeric>
using namespace std;

int main()
{
    forward_list<int> l1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // initilazation
forward_list
    forward_list<int>::iterator it;
    for (auto it = l1.begin(); it != l1.end(); it++)
        cout << *it << " ";

    int sum = 0;
```

```
    for (auto it = l1.begin(); it != l1.end(); it++)
        sum = sum + (*it);

    cout << "\nSum is " << sum << endl;
    return 0;
}
```

```
Output:
1 2 3 4 5 6 7 8 9 10
Sum is 55
```

4. Write a program to reverse forward list elements.

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

int main()
{
    forward_list<int> l1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // initilazation
forward_list
    forward_list<int>::iterator it;
    cout <<"Before reverse" << endl;
    for (auto it = l1.begin(); it != l1.end(); it++)
        cout << *it << " ";
    l1.reverse();
    cout << endl;
    cout <<"After reverse" << endl;
    for (auto it = l1.begin(); it != l1.end(); it++)
        cout << *it << " ";
    return 0;
}
```

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

5. Write a program remove all consecutive duplicate elements from the forward list

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

int main()
{
    forward_list<int> l1;
    l1.assign({1, 1, 2, 3, 5, 5, 6, 7, 9, 10});

    forward_list<int>::iterator it1;
    cout << "Before remove all consecutive duplicate elements from the
forward list" << endl;
    for (auto it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;

    l1.unique();
    cout << "After emove all consecutive duplicate elements from the forward
list" << endl;
    for (it1 = l1.begin(); it1 != l1.end(); it1++)
        cout << *it1 << " ";
    cout << endl;
```

```
    return 0;
}
================================================================================
Output:
Before remove all consecutive duplicate elements from the forward list
1 1 2 3 5 5 6 7 9 10
After emove all consecutive duplicate elements from the forward list
1 2 3 5 6 7 9 10
```

6. Create two forward lists of int type, and merge them.

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

int main()
{
    forward_list<int> list1 = {10, 20, 30, 40, 50};
    forward_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
```

7. Below are two forward lists, first sort them and then merge them.

   forwardlist1={3,2,9}

   forwardlist2={8,1,2}

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

int main()
{
    forward_list<int> list1 = {3,2,9};
    forward_list<int> list2 = {8, 1, 2};

    list1.sort();
    list2.sort();
    // merge operation
    list1.merge(list2);

    cout << "List: ";

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

    return 0;
}
================================================================================
Output:
List: 1 2 2 3 8 9
```

8. Create two forward lists of int type, and swap the elements of both forward lists with each other.

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

int main()
{
    forward_list<int> list1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    forward_list<int> list2 = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};

    cout << "Before swap" << endl;
    cout << "List1: ";
    for (auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";
    cout << "\nList2: ";
    for (auto it2 = list2.begin(); it2 != list2.end(); it2++)
        cout << *it2 << " ";

    list1.swap(list2);
    cout << "\nAfter swap" << endl;
    cout << "List1: ";

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

    cout << "\nList2: ";

    for (auto it2 = list2.begin(); it2 != list2.end(); it2++)
        cout << *it2 << " ";
    return 0;
}
```

```
Output:
Before swap
List1: 1 2 3 4 5 6 7 8 9 10
List2: 10 20 30 40 50 60 70 80 90 100
After swap
List1: 10 20 30 40 50 60 70 80 90 100
List2: 1 2 3 4 5 6 7 8 9 10
```

9. Write a C++ code to demonstrate working of splice_after() in forward list.

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

int main()
{
    // initialising the forward lists
    forward_list<int> list1 = { 10, 20, 30, 40 };
    forward_list<int> list2 = { 4, 9 };

    // splice_after operation performed
    // all elements except the first element in list1 is
    // inserted in list 2 between 4 and 9
    list2.splice_after(list2.begin(), list1, list1.begin(), list1.end());

    cout << "Elements are: " << endl;

    // loop to print the elements of second list
    for (auto it = list2.begin(); it != list2.end(); ++it)
        cout << *it << " ";

    return 0;
}
```

```
Output:
Elements are:
4 20 30 40 9
```

10. Write a program to assign values in forward_list using the values of another list

```cpp
#include <iostream>
#include <forward_list>
#include <array>
using namespace std;

int main()
{
    array <int, 10> list1{1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    forward_list<int> list2;

    cout << "List1: ";
    for (auto it = list1.begin(); it != list1.end(); it++)
        cout << *it << " ";
    cout << endl;

    list2.assign(list1.begin(), list1.end());
    cout << "List2: ";
    for (auto it2 = list2.begin(); it2 != list2.end(); it2++)
        cout << *it2 << " ";
    cout << endl;

    return 0;
}
```

```
================================================================================
Output:
List1: 1 2 3 4 5 6 7 8 9 10
List2: 1 2 3 4 5 6 7 8 9 10
```