Assignment - 39

A Job Ready Bootcamp in C++, DSA and IOT forward\_list

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

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> first;

    forward\_list<int> second;

    forward\_list<int> third;

    first.assign({1,2,3,4,5,6,7,8,9,10});

    third.assign(6, 23);

    first.merge(third);

    forward\_list<int> :: iterator it;

    for(it = first.begin(); it!= first.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    return 0;

}

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.

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> first;

    forward\_list<int> second;

    forward\_list<int> third;

    first.assign({1,2,3,4,5,6,7,8,9,10});

    second.assign(first.begin(), first.end());

    third.assign(6, 23);

    forward\_list<int> :: iterator it;

    for(it = first.begin(); it!= first.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    for(it = second.begin(); it!= second.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

     for(it = third.begin(); it!= third.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    return 0;

}

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

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> first;

    first.assign({1,2,3,4,5,6,7,8,9,10});

    int sum = 0;

    forward\_list<int> :: iterator it;

    for(it = first.begin(); it!= first.end(); it++)

    {

       sum=sum+\*it;

    }

     cout<<"Sum of 1 to 10 is:"<<sum;

    return 0;

}

4. Write a program to reverse forward list elements.

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> first;

    first.assign({1,1,2,3,3,3,4,5,6,7,7,7,7,8,9,10});

    first.reverse();

    forward\_list<int> :: iterator it;

    for(it = first.begin(); it!= first.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

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

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> first;

    first.assign({1,1,2,3,3,3,4,5,6,7,7,7,7,8,9,10});

    first.unique();

    forward\_list<int> :: iterator it;

    for(it = first.begin(); it!= first.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

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

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> first;

    forward\_list<int> second;

    forward\_list<int> third;

    first.assign({1,2,3,4,5,6,7,8,9,10});

    third.assign(6, 23);

    first.merge(third);

    forward\_list<int> :: iterator it;

    for(it = first.begin(); it!= first.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    return 0;

}

7. Below are two forward lists, first sort them and then merge them. forwardlist1={3,2,9} forwardlist2={8,1,2}

//Below are two forward lists, first sort them and then merge them. forwardlist1={3,2,9} forwardlist2={8,1,2}

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> forwardlist1={3,2,9};

    forward\_list<int> forwardlist2={8,1,2};

    forwardlist1.sort();

    forwardlist2.sort();

    forwardlist1.merge(forwardlist2);

    forward\_list<int> :: iterator it;

    for(it = forwardlist1.begin(); it!= forwardlist1.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    return 0;

}

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

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

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> forwardlist1={3,2,9};

    forward\_list<int> forwardlist2={8,1,2};

    forwardlist1.swap(forwardlist2);

    forward\_list<int> :: iterator it;

    for(it = forwardlist1.begin(); it!= forwardlist1.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

     for(it = forwardlist2.begin(); it!= forwardlist2.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    return 0;

}

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

//Write a C++ code to demonstrate working of splice\_after() in forward list.

#include<bits/stdc++.h>

#include<iterator>

#include<cstdlib>

using namespace std;

int main()

{

    forward\_list<int> forwardlist1={3,2,9,4};

    forward\_list<int> forwardlist2={8,1,2,22};

    forward\_list<int>:: const\_iterator it= forwardlist2.begin();

    it++;

    forwardlist2.splice\_after(it, forwardlist1.before\_begin(), forwardlist1.end());

    cout<<"Elements are:"<<endl;

    for (auto it = forwardlist2.begin(); it != forwardlist2.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

10. Write a program to assign values in forward\_list using the values of another list

#include<bits/stdc++.h>

using namespace std;

int main()

{

    forward\_list<int> list1;

    forward\_list<int> list2;

    list1.assign({1,2,3,5});

    list2.assign(list1.begin(), list1.end());

    for(int a : list2)

    {

        cout<<a<<" ";

    }

}