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

**forward\_list**

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

Sol – 1.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl1={5,2,1};

forward\_list<int>fl2={7,1,6};

fl1.sort();

fl2.sort();

fl1.merge(fl2);

cout<<"Elements after merge : "<<endl;

for(auto i=fl1.begin();i!=fl1.end();++i)

cout<<\*i<<" ";

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.

Sol – 2.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl1;

forward\_list<int>fl2;

fl1.assign({1,5,3});

fl2.assign(5,10);

cout<<"Elements of first list are :";

for(int a : fl1)

cout<<a<<" ";

cout<<endl;

cout<<"Elements of second list are :";

for(int a : fl2)

cout<<a<<" ";

cout<<endl;

return 0;

}

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

Sol – 3.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,2,3,4,5,6,7,8,9,10};

int sum=0;

while(!fl.empty())

{

sum=sum+fl.front();

fl.pop\_front();

}

cout<<"Sum : "<<sum;

cout<<endl;

return 0;

}

4. Write a program to reverse forward list elements.

Sol – 4.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,2,3,4,5,6,7,8,9,10};

cout<<"Elements without reverse : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

fl.reverse();

cout<<"Elements after reverse : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

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

Sol – 5.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,2,2,6,7,7,8,8,10};

fl.unique();

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

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

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

Sol – 6.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,2,6,7,8,10};

forward\_list<int>fl2={3,4,5,9};

fl.merge(fl2);

for(auto it=fl.begin();it!=fl.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}

Sol – 7.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,6,7,2,8,10};

forward\_list<int>fl2={5,4,3,9};

fl.sort();

fl2.sort();

fl.merge(fl2);

for(auto it=fl.begin();it!=fl.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.

Sol – 8.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,6,7,2,8,10};

forward\_list<int>fl2={5,4,3,9};

cout<<"Before Swapping"<<endl;

cout<<"Forward List 1 : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

cout<<"Forward List 2 : ";

for(auto it=fl2.begin();it!=fl2.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

fl.swap(fl2);

cout<<"After Swapping"<<endl;

cout<<"Forward List 1 : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

cout<<"Forward List 2 : ";

for(auto it=fl2.begin();it!=fl2.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

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

Sol – 9.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,6,7,2,8,10};

forward\_list<int>fl2={5,4,3,9};

forward\_list<int>::const\_iterator it=fl2.begin();

it++;

fl2.splice\_after(it,fl,fl.before\_begin(),fl.end());

cout<<"Elements are : ";

for(auto it=fl2.begin();it!=fl2.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

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

Sol – 10.

#include<bits/stdc++.h>

using namespace std;

int main()

{

forward\_list<int>fl={1,6,7,2,8,10};

forward\_list<int>fl2;

fl2.assign(fl.begin(),fl.end());

cout<<"Elements are : ";

for(auto it=fl2.begin();it!=fl2.end();it++)

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}