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

**list**

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

Sol – 1.

#include<iostream>

using namespace std;

#include<list>

int main()

{

list<int>l;

if(l.empty())

cout<<"List is empty"<<endl;

else

cout<<"List is not empty"<<endl;

l.push\_front(10);

l.push\_front(20);

l.push\_front(30);

l.push\_front(40);

l.push\_front(50);

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

cout<<endl;

it=l.begin();

l.insert(++it,100);

l.insert(it,200);

l.insert(it,300);

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

{

cout<<\*it<<" ";

}

cout<<endl;

cout<<"Size of list is : "<<l.size()<<endl;

l.sort();

l.reverse();

l.remove(100);

cout<<"First element : "<<l.front()<<endl;

cout<<"Last element : "<<l.back()<<endl;

l.pop\_front();

l.pop\_back();

list<int>::reverse\_iterator rit;

for(rit=l.rbegin();rit!=l.rend();rit++)

{

cout<<\*rit<<" ";

}

cout<<endl;

list<int>l2;

l2.assign(5,100);

l.merge(l2);

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

{

cout<<\*it<<" ";

}

cout<<endl;

l.unique();

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

{

cout<<\*it<<" ";

}

cout<<endl;

it=l.begin();

l.erase(it);

return 0;

}

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

STL

Sol – 2.

#include<iostream>

using namespace std;

#include<list>

void display(list<int>l)

{

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

cout<<endl;

}

int main()

{

list<int>l1;

list<int>l2;

list<int>l3;

list<int>l4;

l1.assign(5,100);

display(l1);

l2.assign(l1.begin(),l1.end());

display(l2);

int a[]={10,20,30,40};

l3.assign(a,a+4);

display(l3);

return 0;

}

3. Iterate a list C++ STL

Sol – 3.

Same as 1

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

Sol – 4.

#include<bits/stdc++.h>

using namespace std;

int main()

{

list<int>l={1,5,9,4,6};

list<int>::iterator it;

for(it=l.end();it!=l.begin();)

{

it--;

cout<<\*it<<" ";

}

cout<<endl;

l.reverse();

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

{

cout<<\*it<<" ";

}

l.reverse();

cout<<endl;

list<int>::reverse\_iterator rit;

for(rit=l.rbegin();rit!=l.rend();rit++)

{

cout<<\*rit<<" ";

}

return 0;

}

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

Sol – 5.

#include<bits/stdc++.h>

using namespace std;

int main()

{

list<string>l;

string str;

while(true)

{

cout<<"Enter string(\"EXIT\" to quit):";

getline(cin,str);

if(str=="EXIT")

break;

l.push\_back(str);

}

cout<<"List elements are : "<<endl;

list<string>::iterator it;

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

{

cout<<\*it<<" ";

}

return 0;

}

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

Sol – 6.

#include<bits/stdc++.h>

using namespace std;

int main()

{

list<int>l={2,6,3,4,5};

cout<<"First element : "<<l.front()<<endl;

cout<<"Last element : "<<l.back()<<endl;

return 0;

}

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

Sol – 7.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

list<int>l={2,6,3,4,5};

cout<<"Enter an element to insert at front : ";

cin>>n;

l.push\_front(n);

cout<<"Enter an element to insert at front : ";

cin>>n;

l.push\_back(n);

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

return 0;

}

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

C++ STL

Sol – 8.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

list<int>l={2,6,3,4,5,2,7,1};

l.remove(2);

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

return 0;

}

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

Sol – 9.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

list<int>l={2,6,3,3,4,5,5,2,7,7,1};

l.unique();

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

return 0;

}

10. Merge two lists C++ STL

Sol – 10.

#include<bits/stdc++.h>

using namespace std;

void printlist(list<int>l)

{

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

cout<<endl;

}

int main()

{

int n;

list<int>l1={2,6,7,1};

printlist(l1);

list<int>l2={5,4,9,3};

printlist(l2);

l1.merge(l2);

cout<<"After merging : ";

printlist(l1);

return 0;

}

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

Sol – 11.

#include<bits/stdc++.h>

using namespace std;

void printlist(list<int>l)

{

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

cout<<endl;

}

int main()

{

int n;

list<int>l1={2,6,7,1};

printlist(l1);

list<int>l2;

l2.assign(l1.begin(),l1.end());

printlist(l2);

return 0;

}

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

Sol – 12.

#include<bits/stdc++.h>

using namespace std;

void printlist(list<int>l)

{

list<int>::iterator it;

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

{

cout<<\*it<<" ";

}

cout<<endl;

}

int main()

{

int n[]={1,5,4,7,6,9};

list<int>l1;

l1.assign(n,n+6);

printlist(l1);

return 0;

}

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

Sol – 13.

#include<bits/stdc++.h>

using namespace std;

int main()

{

list<char>l1;

l1.push\_back('k');

l1.push\_back('l');

l1.push\_back('z');

l1.push\_back('r');

l1.push\_back('u');

for(char x: l1)

{

cout<<x<<" ";

}

cout<<endl;

return 0;

}

14. Access elements of a characters list using const\_iterator in C++ STL

Sol – 14.

#include<bits/stdc++.h>

using namespace std;

void printlist(list<char>l)

{

list<char>::const\_iterator it;

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

{

cout<<\*it<<" ";

}

cout<<endl;

}

int main()

{

list<char>l1;

l1.push\_back('k');

l1.push\_back('l');

l1.push\_back('z');

l1.push\_back('r');

l1.push\_back('u');

printlist(l1);

return 0;

}