# 15B17CI371 – Data Structures Lab
# ODD 2024
# Week 6-LAB A
# Practice Lab - STL

**1. Use vectors to apply sorting to any array.**

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
   int n,num;
cout<<"Input the size of the array : ";
cin>>n;
   vector<int>arr;
cout<<"Input the elements : ";
   for(int i=0;i<n;i++)
   {
cin>>num;
arr.push_back(num);
   }
   for(int i=0;i<n;i++)
      for(int j=0;j<n-i-1;j++)
         if(arr[j]>arr[j+1])
         {
            int temp=arr[j];
arr[j]=arr[j+1];
arr[j+1]=temp;
         }
cout<<"Sorted Array : ";
   for (int i:arr)
cout<<i<< " ";
}
```

**Output :**

```
Input the size of the array : 7
Input the elements : 4
2
6
1
5
7
3
Sorted Array : 1 2 3 4 5 6 7
Process returned 0 (0x0)   execution time : 10.293 s
Press any key to continue.
```

**2. Use STL to :-**

**a. count the frequency of a particular value in a given array.**

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main()
{
   int n,num,val;
cout<<"Input the size of the array : ";
cin>>n;
   vector<int>arr;
cout<<"Input the elements : ";
   for(int i=0;i<n;i++)
   {
cin>>num;
arr.push_back(num);
   }
cout<<"Input the element to find out its frequency : ";
cin>>val;
   int frequency=count(arr.begin(),arr.end(),val);
cout<< "Frequency of " <<val<< ": "<<frequency;
}
```

**Output :**

```
Input the size of the array : 6
Input the elements : 3
1
3
4
5
2
Input the element to find out its frequency : 3
Frequency of 3: 2
Process returned 0 (0x0)   execution time : 7.892 s
Press any key to continue.
```

**b. erase a selected element in vector,shift and resizes the vector elements accordingly (after deletion of the selected element).**

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main()
{
   int n,num,ind;
cout<<"Input the size of the array : ";
cin>>n;
   vector<int>arr;
cout<<"Input the elements : ";
   for(int i=0;i<n;i++)
   {
cin>>num;
arr.push_back(num);
   }
cout<<"Input the index whose element is to be deleted : ";
cin>>ind;
arr.erase(arr.begin() + ind);
cout<<"New array : ";
   for (int i:arr)
cout<<i<< " ";
}
```

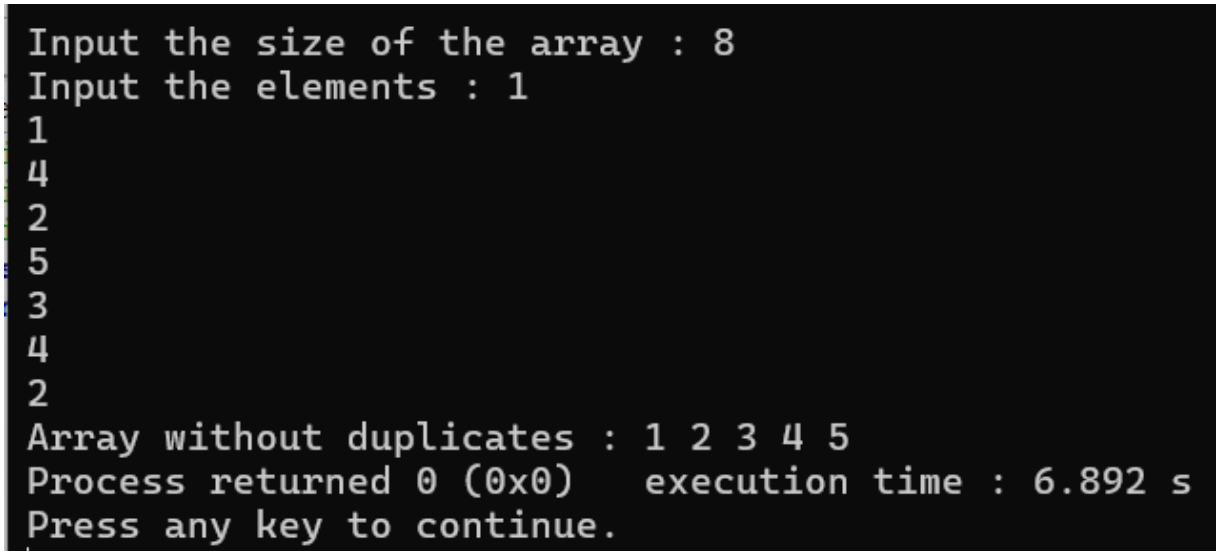
**Output :**

```
Input the size of the array : 6
Input the elements : 1
2
3
3
4
5
Input the index whose element is to be deleted : 2
New array : 1 2 3 4 5
Process returned 0 (0x0)   execution time : 7.900 s
Press any key to continue.
```

**c. erase duplicates in a given vector.**

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main()
{
   int n,num,ind;
cout<<"Input the size of the array : ";
cin>>n;
   vector<int>arr;
cout<<"Input the elements : ";
   for(int i=0;i<n;i++)
   {
cin>>num;
arr.push_back(num);
   }
   sort(arr.begin(),arr.end());
arr.erase(unique(arr.begin(),arr.end()),arr.end());
cout<<"Array without duplicates : ";
   for (int i:arr)
cout<<i<< " ";
}
```

**Output :**




**d. find the distance between the first element and the maximum value  within an array**

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main()
{
   int n,num,ind;
cout<<"Input the size of the array : ";
cin>>n;
   vector<int>arr;
cout<<"Input the elements : ";
   for(int i=0;i<n;i++)
   {
cin>>num;
arr.push_back(num);
   }
   int dis=distance(arr.begin(),max_element(arr.begin(),arr.end()));
cout<< "Distance between first element and max element: "<<dis;
}
```

**Output :**

```
Input the size of the array : 7
Input the elements : 2
3
1
4
5
9
6
Distance between first element and max element: 5
Process returned 0 (0x0)   execution time : 20.224 s
Press any key to continue.
```

**3. Use std::list (class of the List container) to perform the following:**
**a. Finds the value of the first element in the list.**
**b. Finds the value of the last element in the list.**
**c. Adds a new element at the end of the list.**
**d. Removes the first element of the list,and reduces the size of the listby1.**
**e. Inserts new elements in the list before the element at a specifiedposition.**
**f. Returns the size of the list.**
**g. Removes all the elements from the list,which are equal to a givenelement.**
**h. Reverses the list.**
**i. Removes all duplicate consecutive elements from the list.**
**j. swap the contents of one list with another list.**

```
#include <iostream>
#include <list>
using namespace std;
int main()
{
   int n,num,pos;
   list<int> l;
   cout<<"Input the number of elements : ";
   cin>>n;
   cout<<"Input the elements : ";
   for(int i=0;i<n;i++)
   {
      cin>>num;
      l.push_front(num);
   }
   cout<<"List : ";
   for(int i:l)
      cout<<i<<" ";
   cout<<endl<<"First element of the list : "<<l.front() <<endl;
   cout<<"Last element of the list : "<<l.back() <<endl;
   cout<<"Input the element to add at the end of the list : ";
   cin>>num;
   l.push_back(num);
   cout<<"Updated List : ";
   for(int i:l)
      cout<<i<<" ";
   cout<<endl<<"Removing first element of the list :"<<endl;
```

```
l.pop_front();
cout<<"Updated List : ";
for(int i:l)
    cout<<i<<" ";
cout<<endl<<"Input the element to add at the specified index of the list : ";
cin>>num;
cout<<"Input the index : ";
cin>>pos;
auto it=l.begin();
advance(it,pos);
l.insert(it,num);
cout<<"Updated List : ";
for(int i:l)
    cout<<i<<" ";
size_t listSize=l.size();
cout<<endl<<"Size of the list : "<<listSize<<endl;
cout<<"Input the element to be removed from the list : ";
cin>>num;
l.remove(num);
cout<<"Updated List : ";
for(int i:l)
    cout<<i<<" ";
cout<<endl<<"Reversing the list :"<<endl;
l.reverse();
cout<<"Updated List : ";
for(int i:l)
    cout<<i<<" ";
cout<<endl<<"Removing consecutive duplicate elements :"<<endl;
l.unique();
cout<<"Updated List : ";
for(int i:l)
    cout<<i<<" ";
    cout<<"Another List :"<<endl;
list<int> l2;
cout<<"Input the number of elements : ";
cin>>n;
cout<<"Input the elements : ";
for(int i=0;i<n;i++)
{
    cin>>num;
    l.push_front(num);
}
cout<<"Before Swapping :\nList 1: ";
for(int i:l)
    cout<<i<<" ";
cout<<"List 2: ";
for(int i:l2)
    cout<<i<<" ";
l.swap(l2);
cout<<"After Swapping :\nList 1: ";
for(int i:l)
    cout<<i<<" ";
cout<<"List 2: ";
for(int i:l2)
    cout<<i<<" ";
```

}

**Output :**

```
Input the elements : 1
2
3
4
5
5
6
7
List : 7 6 5 5 4 3 2 1
First element of the list : 7
Last element of the list : 1
Input the element to add at the end of the list : 0
Updated List : 7 6 5 5 4 3 2 1 0
Removing first element of the list :
Updated List : 6 5 5 4 3 2 1 0
Input the element to add at the specified index of the list : 4
Input the index : 3
Updated List : 6 5 5 4 4 3 2 1 0
Size of the list : 9
Input the element to be removed from the list : 0
Updated List : 6 5 5 4 4 3 2 1
Reversing the list :
Updated List : 1 2 3 4 4 5 5 6
Removing consecutive duplicate elements :
Updated List : 1 2 3 4 5 6
Another List :
Input the number of elements : 6
Input the elements : 0
9
8
7
6
5
Before Swapping :
List 1: 1 2 3 4 5 6
List 2: 5 6 7 8 9 0
After Swapping :
List 1: 5 6 7 8 9 0
List 2: 1 2 3 4 5 6
Process returned 0 (0x0)   execution time : 37.892 s
Press any key to continue.
```

**4. Use std::map Member Functions to**
**a. Find the number of elements in the map.**
**b. Add a new element to the map.**
**c. Removes the key-value ‘g’ from the map.**

#include <iostream>

```
#include <map>
using namespace std;
int main()
{
   map<char,int> M;
   M['a']=1;
   M['b']=2;
   M['c']=3;
   M['g']=7;
   cout<<"Contents of the map :"<<endl;
   for(auto it:M)
      cout<<it.first<<"=>"<<it.second<<endl;
   cout<<"Number of elements: "<<M.size()<<endl<<endl;
   M['d']=4;
   cout<<"Added element 'd'."<<endl;
   cout<<"Contents of the map :"<<endl;
   for(auto it:M)
      cout<<it.first<<"=>"<<it.second<<endl;
   cout<<"Number of elements: "<<M.size()<<endl<<endl;
   M.erase('g');
   cout<<"Removing key 'g'."<<endl;
   cout<<"Contents of the map :"<<endl;
   for(auto it:M)
      cout<<it.first<<"=>"<<it.second<<endl;
   cout<<"Number of elements: "<<M.size();
}
```

**Output :**

```
Contents of the map :
a=>1
b=>2
c=>3
g=>7
Number of elements: 4

Added element 'd'.
Contents of the map :
a=>1
b=>2
c=>3
d=>4
g=>7
Number of elements: 5

Removing key 'g'.
Contents of the map :
a=>1
b=>2
c=>3
d=>4
Number of elements: 4
Process returned 0 (0x0)   execution time : 0.092 s
Press any key to continue.
```