15B17CI371 – Data Structures Lab

ODD 2024

9923103023 – F1

Week 6-LAB A

Practice Lab – STL

Lab Questions:

1. Use vectors to apply sorting to any array.

Ans : #include <iostream>

#include <vector>

using namespace std;

void sort\_arr(vector<int> &*arr*)

{

    for (int i = 0; i < *arr*.size(); i++)

    {

        for (int j = i + 1; j < *arr*.size(); j++)

        {

            if (*arr*[i] > *arr*[j])

            {

                swap(*arr*[i], *arr*[j]);

            }

        }

    }

}

int main()

{

    vector<int> v;

    int n;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the unsorted array : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        v.push\_back(temp);

    }

    cout << "the sorted array is : ";

    sort\_arr(v);

    for (int i : v)

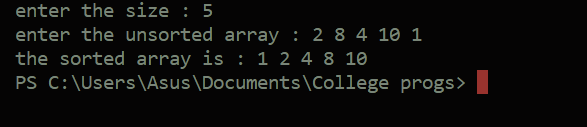
    {

        cout << i << " ";

    }

    return 0;

}



Q2 Use STL to

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

Ans :

#include <iostream>

#include <vector>

using namespace std;

void count\_freq(vector<int> &*arr*)

{

    vector<int> freq(10, 0);

    for (int i = 0; i < *arr*.size(); i++)

    {

        freq[*arr*[i]]++;

    }

    for (int i = 0; i < 10; i++)

    {

        if (freq[i] > 0)

        {

            cout << "frequency of " << i << "is : " << freq[i] << " \n";

        }

        else

        {

            continue;

        }

    }

}

int main()

{

    vector<int> v;

    int n;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the array : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

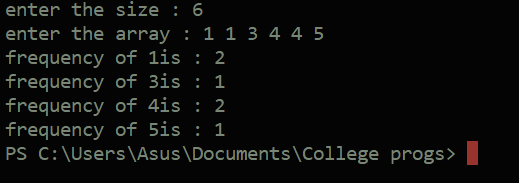
        v.push\_back(temp);

    }

    count\_freq(v);

    return 0;

}



b. erase a selected element in vector, shift and resizes the vector elements

accordingly (after deletion of the selected element).

Ans :

#include <iostream>

#include <vector>

using namespace std;

void del(vector<int> &*arr*, int *selec*)

{

    int index;

    for (int i = 0; i < *arr*.size(); i++)

    {

        if (*arr*[i] == *selec*)

        {

            index = i;

            break;

        }

    }

    for (int i = index; i < *arr*.size(); i++)

    {

*arr*[i] = *arr*[i + 1];

    }

*arr*.pop\_back();

}

int main()

{

    vector<int> v;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the array : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        v.push\_back(temp);

    }

    cout << "enter the target : ";

    cin >> target;

    del(v, target);

    for (int i : v)

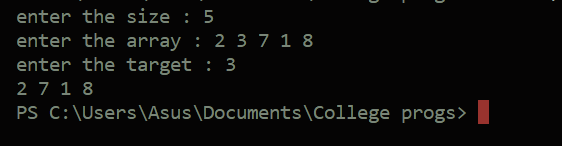
    {

        cout << i << " ";

    }

    return 0;

}



c. erase duplicates in a given vector.

Ans : #include <iostream>

#include <vector>

using namespace std;

void del(vector<int> &*arr*, int *selec*, int *start*)

{

    int index;

    for (int i = *start*; i < *arr*.size(); i++)

    {

        if (*arr*[i] == *selec*)

        {

            index = i;

            break;

        }

    }

    for (int i = index; i < *arr*.size(); i++)

    {

*arr*[i] = *arr*[i + 1];

    }

*arr*.pop\_back();

}

void remove\_dupl(vector<int> &*arr*)

{

    for (int i = 0; i < *arr*.size(); i++)

    {

        for (int j = i + 1; j < *arr*.size(); j++)

        {

            if (*arr*[i] == *arr*[j])

            {

                del(*arr*, *arr*[i], j);

            }

        }

    }

}

int main()

{

    vector<int> v;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the array : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        v.push\_back(temp);

    }

    remove\_dupl(v);

    for (int i : v)

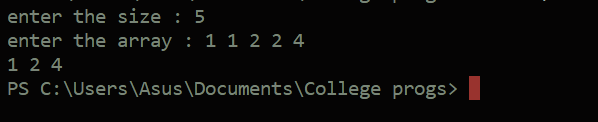
    {

        cout << i << " ";

    }

    return 0;

}



d. find the distance between the first element and the maximum value

within an array

Ans:

#include <iostream>

#include <vector>

using namespace std;

int max\_index(vector<int> &*arr*)

{

    int m = *arr*[0];

    for (int i = 1; i < *arr*.size(); i++)

    {

        if (*arr*[i] > m)

        {

            m = *arr*[i];

        }

    }

    for (int i = 0; i < *arr*.size(); i++)

    {

        if (*arr*[i] == m)

        {

            return i;

        }

    }

}

int main()

{

    vector<int> v;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the array : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

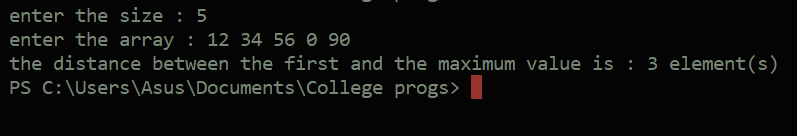
        v.push\_back(temp);

    }

    cout << "the distance between the first and the maximum value is : " << max\_index(v) - 1 << " element(s) ";

    return 0;

}



Q3. Use std::list (class of the List container) to perform the following:

1. Finds the value of the first element in the list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

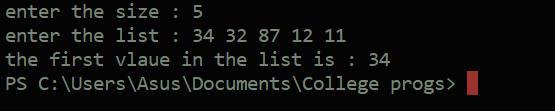
        lis.push\_back(temp);

    }

    cout << "the first vlaue in the list is : " << lis.front();

    return 0;

}



1. Finds the value of the last element in the list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

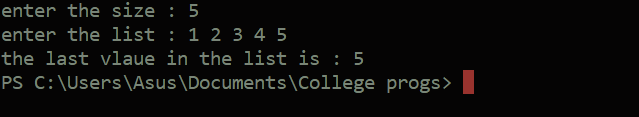
        lis.push\_back(temp);

    }

    cout << "the last vlaue in the list is : " << lis.back();

    return 0;

}



1. Adds a new element at the end of the list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        lis.push\_back(temp);

    }

    cout << "the list is : ";

    for (int i : lis)

    {

        cout << i << " ";

    }

    cout << "\n";

    cout << "the list after adding 6 at the end : ";

    lis.push\_back(6);

    for (int i : lis)

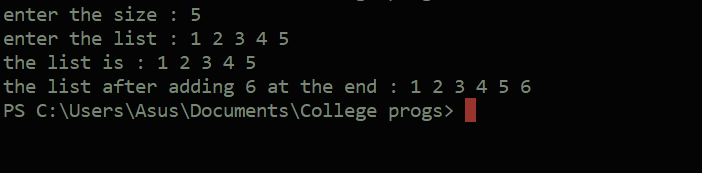
    {

        cout << i << " ";

    }

    return 0;

}



d. Removes the first element of the list, and reduces the size of the list

by 1.

Ans : #include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        lis.push\_back(temp);

    }

    cout << "the list is : ";

    for (int i : lis)

    {

        cout << i << " ";

    }

    cout << "\n";

    cout << "the list after removing the first element :  ";

    lis.pop\_front();

    for (int i : lis)

    {

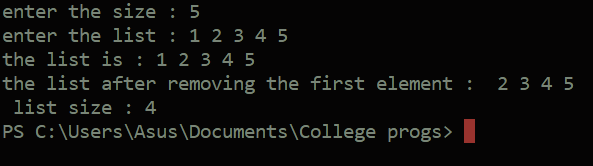
        cout << i << " ";

    }

    cout << "\n list size : " << lis.size();

    return 0;

}



e. Inserts new elements in the list before the element at a specified

position.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, add, pos;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        lis.push\_back(temp);

    }

    cout << "enter the element to add and the position to add : ";

    cin >> add >> pos;

    auto it = lis.begin();

    advance(it, pos - 1);

    lis.insert(it, add);

    cout << "list after insertion : ";

    for (int i : lis)

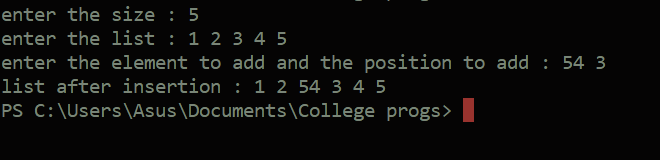
    {

        cout << i << " ";

    }

    return 0;

}



f. Returns the size of the list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, target;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

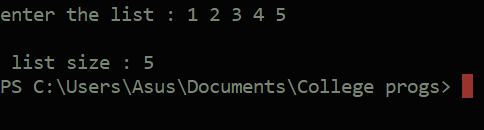
        lis.push\_back(temp);

    }

    cout << "\n list size : " << lis.size();

    return 0;

}



g. Removes all the elements from the list, which are equal to a given

element.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, add, pos;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        lis.push\_back(temp);

    }

    cout << "enter the element to be removed : ";

    cin >> add;

    lis.remove(add);

    cout << "list after removal : ";

    for (int i : lis)

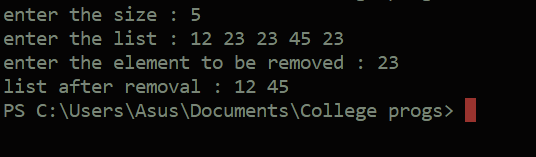
    {

        cout << i << " ";

    }

    return 0;

}



h. Reverses the list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, add, pos;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        lis.push\_back(temp);

    }

    lis.reverse();

    cout << "list after reversal : ";

    for (int i : lis)

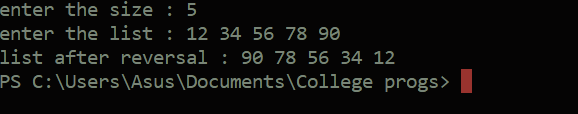
    {

        cout << i << " ";

    }

    return 0;

}



i. Removes all duplicate consecutive elements from the list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> lis;

    int n, add, pos;

    cout << "enter the size : ";

    cin >> n;

    cout << "enter the list : ";

    for (int i = 0; i < n; i++)

    {

        int temp;

        cin >> temp;

        lis.push\_back(temp);

    }

    lis.unique();

    cout << "list after removing consecutive duplicates : ";

    for (int i : lis)

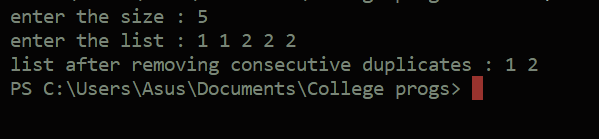
    {

        cout << i << " ";

    }

    return 0;

}



j. swap the contents of one list with another list.

Ans :

#include <iostream>

#include <list>

using namespace std;

int main()

{

    list<int> list1 = {1, 2, 3, 4, 12};

    list<int> list2 = {8, 5, 4, 72, 9, 4};

    cout << "before swapping : \n";

    cout << "list 1 :";

    for (int i : list1)

    {

        cout << i << " ";

    }

    cout << "\n list2 : ";

    for (int i : list2)

    {

        cout << i << " ";

    }

    list1.swap(list2);

    cout << "\nlist after swaping : \n";

    cout << "list 1 :";

    for (int i : list1)

    {

        cout << i << " ";

    }

    cout << "\n list2 : ";

    for (int i : list2)

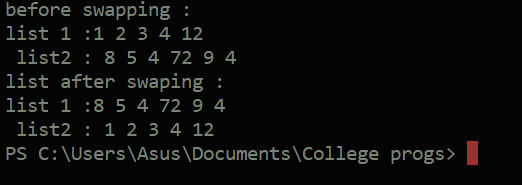
    {

        cout << i << " ";

    }

    return 0;

}



Q4. Use std::map Member Functions to

1. Find the number of elements in the map.

Ans :

#include <iostream>

#include <map>

using namespace std;

int main()

{

    map<int, string> myMap;

    myMap[1] = "Apple";

    myMap[2] = "Banana";

    myMap[3] = "Cherry";

    cout << "Map contents:" << endl;

    for (const auto &pair : myMap)

    {

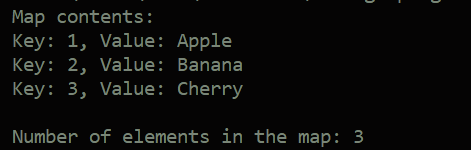
        cout << "Key: " << pair.first << ", Value: " << pair.second << endl;

    }

    cout << "\nNumber of elements in the map: " << myMap.size() << endl;

    return 0;

}



1. Add a new element to the map.

Ans :

#include <iostream>

#include <map>

using namespace std;

int main()

{

    map<int, string> myMap;

    myMap[1] = "Apple";

    myMap[2] = "Banana";

    myMap[3] = "Cherry";

    cout << "Map contents before adding an element :" << endl;

    for (const auto &pair : myMap)

    {

        cout << "Key: " << pair.first << ", Value: " << pair.second << endl;

    }

    myMap[4] = "newAdded";

    cout << "Map contents after adding an element :" << endl;

    for (const auto &pair : myMap)

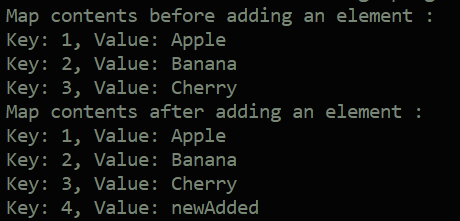
    {

        cout << "Key: " << pair.first << ", Value: " << pair.second << endl;

    }

    return 0;

}



1. Removes the key-value ‘g’ from the map.

Ans :

#include <iostream>

#include <map>

using namespace std;

int main()

{

    map<char, string> myMap;

    myMap['a'] = "Apple";

    myMap['b'] = "Banana";

    myMap['c'] = "Cherry";

    myMap['g'] = "Grapes";

    myMap['m'] = "Mango";

    cout << "Map contents before removing 'g':" << endl;

    for (const auto &pair : myMap)

    {

        cout << "Key: " << pair.first << ", Value: " << pair.second << endl;

    }

    myMap.erase('g');

    cout << "\nMap contents after removing 'g':" << endl;

    for (const auto &pair : myMap)

    {

        cout << "Key: " << pair.first << ", Value: " << pair.second << endl;

    }

    return 0;}

}