Assignment – 36

A Job Ready Bootcamp in C++, DSA and IOT

STL: array

1. Using STL Array gets and sets a reference to an element based on a given index.

#include<array>

#include<iostream>

using namespace std;

int main()

{

    array<int,5> arr {12, 3, 45, 34, 556};

    array<char,7> charbrr {'s','a','t','i','s','h'};

    //print all elements of the array

    cout<<"Elements at index 0 :"<<arr[0]<<endl;

    cout<<"Elements at index 1 :"<<arr[1]<<endl;

    cout<<"Elements at index 2 :"<<arr[2]<<endl;

    cout<<"Elements at index 3 :"<<arr[3]<<endl;

    cout<<"Elements at index 4 :"<<arr[4]<<endl;

    arr[1]=100;

    arr[4]=50;

    cout<<endl;

    for(char c : charbrr)

    {

        cout<<c<<" ";

    }

    cout<<endl;

    for(int i : arr)

    {

        cout<<i<<" ";

    }

    cout<<endl;

    return 0;

}

2. Using STL Array returns the total number of elements in the array.

#include<iostream>

#include<array>

using namespace std;

int main()

{

    array<int,10> arr {12,23,34,56,67};

    int count=0;

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

    {

        count++;

    }

    cout<<count<<endl;

    return 0;

}

//Using STL Array returns the total number of elements in the array.

#include<iostream>

#include<array>

using namespace std;

int main()

{

    array<int,10> arr {12,23,34,56,67};

    int count=0;

   for(auto i : arr)

   {

    count++;

   }

    cout<<count<<endl;

    return 0;

}

//Using STL Array returns the total number of elements in the array.

3. Find the first and last element using the STL array.

#include<iostream>

#include<array>

using namespace std;

int main()

{

    array<int,5> arr {12,23,34,56,67};

     cout<<"The first Elements of the given array is :"<<arr[0];

    cout<<" "<<endl;

    cout<<"The last Elements of the given array is :"<<arr[arr.size()-1];

    cout<<" ---------------------------"<<endl;

    cout<<"The first Elements of the given array is :"<<arr.front();

    cout<<" "<<endl;

    cout<<"The last Elements of the given array is :"<<arr.back();

    return 0;

}

//Using STL Array returns the total number of elements in the array.

4. Returns the element from the given index using the STL array.

#include <iostream>

#include <array>

using namespace std;

//without tuple header file included how this get funtion run

int main()

{

    array <int, 5> arr{12,23,13,32,22};

    cout<<arr.at(0);

    cout<<endl;

    cout<<arr.at(1);

    cout<<endl;

    cout<<arr.at(2);

    cout<<endl;

    cout<<get<3>(arr);

    cout<<endl;

    cout<<arr.at(4);

    cout<<endl;

    return 0;

}

5. C++ STL program to demonstrate example of array::rbegin() and array::rend() functions

//C++ STL program to demonstrate example of array::rbegin() and array::rend() functions

#include<iostream>

#include<array>

#include<string>

using namespace std;

int main()

{

    array<int,5> arr {1,2,3,4,5};

    array<string,4> crr {"BHOPAL", "INDORE", "UJJAIN", "JABALPUR"};

    cout<<"Elements of arr array"<<endl;

    for ( auto it=arr.rend()-1; it != arr.rbegin(); it--)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    cout<<"Elements of crr array"<<endl;

    for(auto it=crr.rbegin(); it != crr.rend(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

6. Using STL to check whether an array is empty or not.

#include<iostream>

#include<array>

using namespace std;

int main()

{

    array<int, 0> arr1{};

    array<int, 2> arr2{};

    array<int, 3> arr3{1,2,3};

    cout<<"array.arr1  ---"<<arr1.empty();

    cout<<endl;

    cout<<"array.arr2  ---"<<arr2.empty();

    cout<<endl;

    cout<<"Array.arr3  ---"<<arr3.empty();

    cout<<endl;

    if (arr1.empty())

    {

        cout<<"this is empty array"<<endl;

    }

    else

    cout<<"this is NOT empty array"<<endl;

    if (arr2.empty())

    {

        cout<<"this is empty array"<<endl;

    }

     cout<<"this is NOT empty array"<<endl;

    if (arr3.empty())

    {

        cout<<"this is empty array"<<endl;

    }

    cout<<"this is NOT empty array"<<endl;

    return 0;

}

7. Sort an array in ascending order using sort() function in C++ STL

#include<iostream>

#include<algorithm>

using namespace std;

int main()

{

    int arr[]={12,32,2,34,4};

    int size = sizeof(arr)/sizeof(int);

    //here we wrote (arr+4) because it works as itrator so we will have add 1 extra

    sort(arr, arr+5);

    for (int i = 0; i < size; i++)

    {

       cout<<arr[i]<<" ";

    }

}

8. Sort an array in descending order using sort() function in C++ STL

#include<iostream>

#include<algorithm>

using namespace std;

int main()

{

    int arr[]={12,32,2,34,4};

    int size = sizeof(arr)/sizeof(int);

    //here we wrote (arr+4) because it works as itrator so we will have add 1 extra

    sort(arr, arr+5, greater<>());

    for (int i = 0; i < size; i++)

    {

       cout<<arr[i]<<" ";

    }

}

9. C++ program to find the integers which come an odd number of times in an array using C++ STL.

#include<iostream>

#include<array>

using namespace std;

//The best method for time complexity and space complexity

int odd\_unique(array<int,7> arr)

{

    int result = 0;

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

    {

        result = result^arr[i];

    }

    return result;

}

int main()

{

    array<int,7> arr{1,2,3,1,2,4,4};

    int res= odd\_unique(arr);

    cout<<res;

}

10. Given an integer array nums , return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i] .