Name: _______Hong Hai Dang Nguyen_________ Student ID:___103503191____

SWE20004 Technical Software Development

Lab 6 (week 6)

You will need:
Online C++ IDE or Computer installed IDE
A computer with internet access
A computer with internet access

In this lab you will investigate C++ Arrays and Vectors

Before you start the lab exercise, download an appropriate C++ IDE to run your commands and programs.

1. Answer the following questions? Don't copy and paste – write down what it is - in your own words.

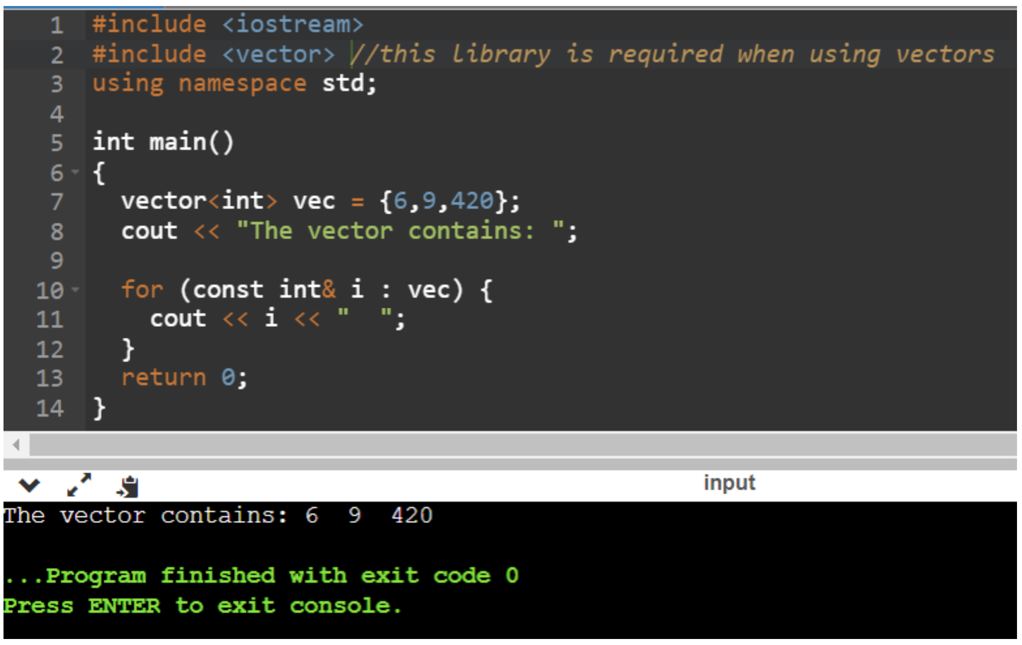
1.1 What is an Array? Give an example

An array is used to hold many values in a single variable instead of creating distinct variables for each value. An array can be declared by indicating the array's name in square brackets, the type of variable to be used, and the amount of elements it should hold. To insert values, enclose the values in curly brackets enclosed by commas. For example:

```
string phone_brands[3] = {"Apple", "Samsung", "Nokia"};
```

1.2 What is a vector? Give an example

A vector is identical to dynamic array but with the ability to resize themselves dynamically when an element is inserted or deleted, with the container handling their storage automatically. The elements of a vector are placed in contiguous storage so they can be retrieved and traversed using iterators. For example:



```
#include <iostream>
#include <vector> //this library is required when using vectors
using namespace std;

int main()
{
  vector<int> vec = {6,9,420};
  cout << "The vector contains: ";

  for (const int& i : vec) {
    cout << i << "  ";
  }
  return 0;
}
```

input

```
The vector contains: 6  9  420

...Program finished with exit code 0
Press ENTER to exit console.
```

Name: _______Hong Hai Dang Nguyen_________ Student ID:___103503191____

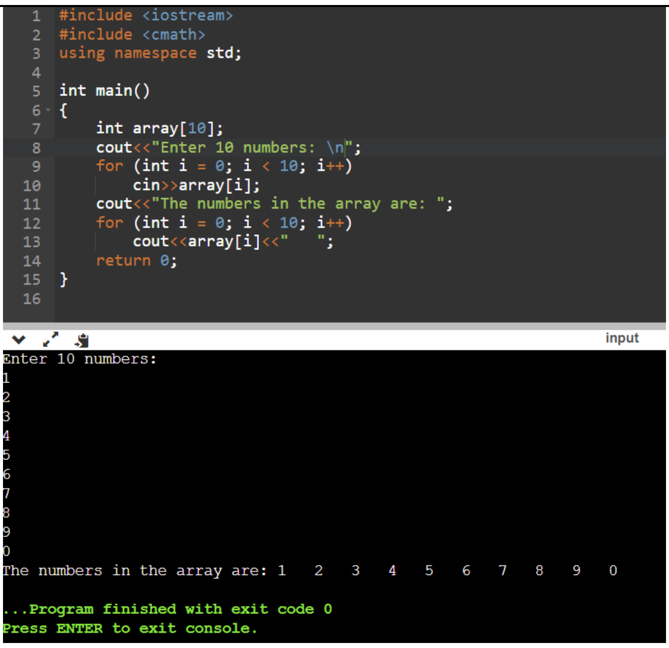
1.3 How can you store 10 inputs in an array? Give an example with a combination of float, char, int

There could only be one type of input for each array. Therefore, we could store 10 inputs in an array with different types of input like below:

```
int arr_int[4]={5, 10, 15, 20};
float arr_float[4]={2.2, 4.4, 6.6, 8.8};
char arr_char[4]={‘l’, ‘m’, ‘a’, ‘o’};
```

1.4 Take 10 integer inputs from user and store them in a array and print them on screen

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int array[10];
    cout<<"Enter 10 numbers: \n";
    for (int i = 0; i < 10; i++)
        cin>>array[i];
    cout<<"The numbers in the array are: ";
    for (int i = 0; i < 10; i++)
        cout<<array[i]<<"   ";
    return 0;
}
```



```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int array[10];
    cout<<"Enter 10 numbers: \n";
    for (int i = 0; i < 10; i++)
        cin>>array[i];
    cout<<"The numbers in the array are: ";
    for (int i = 0; i < 10; i++)
        cout<<array[i]<<"   ";
    return 0;
}
```

```
Enter 10 numbers:
1
2
3
4
5
6
7
8
9
0
The numbers in the array are: 1   2   3   4   5   6   7   8   9   0

...Program finished with exit code 0
Press ENTER to exit console.
```

Name: _______Hong Hai Dang Nguyen_________ Student ID:___103503191____

2. Find the error in the following by writing it in a program and write the program without any errors in the below box.

The errors are in red inside a (), and the fixed codes are in blue

a) #include <iostream>; (Error: Semicolon at the end. Fix: #include<iostream>)

b) arraySize = 10; // arraySize was declared const (Error: Uninitialized const 'arraySize', wrong assignment of read-only variable 'arraySize'. Fix: const int arraySize=10;)

c) Assume that int b[10] = {};
for (int i = 0; i <= 10; i++) b[i] = 1;
(Logical error: The for loop range has to be from 0 to 9 or less than 10. Fix: for(int i=0;i<10;i++))

d) Assume that int a[2][2] = { { 1, 2 }, { 3, 4 } }; a[1, 1] = 5; (Error: incompatible types in assignment of 'int' to 'int [2]'. Fix: a[1][1]=5;)

e)
```
double cube(float );  /*function prototype*/
...
    cube(float number ) /*function prototype*/
    {
    return number*number *number;
    }
```
(error: ambiguating new declaration of 'int cube(float)'. Fix:
```
double cube(float number)
{
return number*number*number;
}
```
)

f)
```
double y =123.45678; int x;
x = y; cout<<(double)x;
```
(Error: there are 2 different data types, double and int don't work together. Fix:
```
double y =123.45678; double x;
x = y; cout<<x;
```
)

g)
```
double square(double number )
{
double number;
return number *number;
}
```
(Error: declaration of 'double number' shadows a parameter. Fix:
```
double square(double number)
{
return number*number;
}
```
)

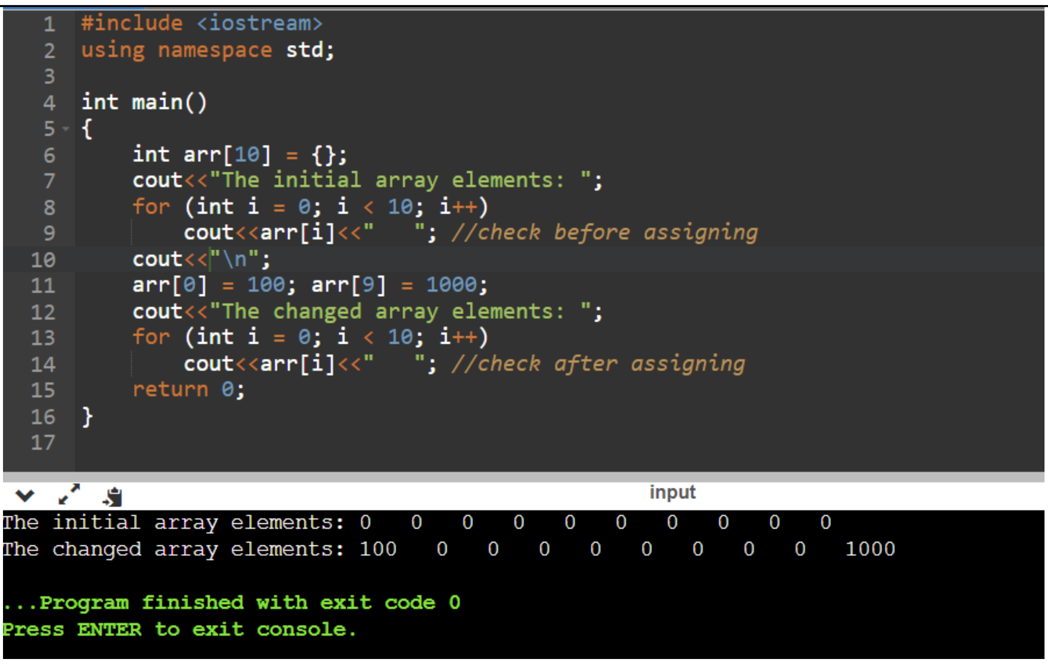
h) double f[3] = { 1.1, 10.01, 100.001, 1000.0001 };
(Error: too many initializers for 'double [3]'. Fix:
double f[]={1.1,10.01,100.001,1000.0001};)

3. Declaring, Initializing and Accessing an array

Name: _______Hong Hai Dang Nguyen_________ Student ID:___103503191____

Declare an array of 10 integers named arr and initialize the array so that all 10 integers are 0.
Then assign 100 to the first element of the array and 1000 to the last element of the array. Try yourself write a program and run it.

```
#include <iostream>
using namespace std;

int main()
{
   int arr[10] = {};
   cout<<"The initial array elements: ";
   for (int i = 0; i < 10; i++)
      cout<<arr[i]<<"   "; //check before assigning
   cout<<"\n";
   arr[0] = 100; arr[9] = 1000;
   cout<<"The changed array elements: ";
   for (int i = 0; i < 10; i++)
      cout<<arr[i]<<"   "; //check after assigning
   return 0;
}
```



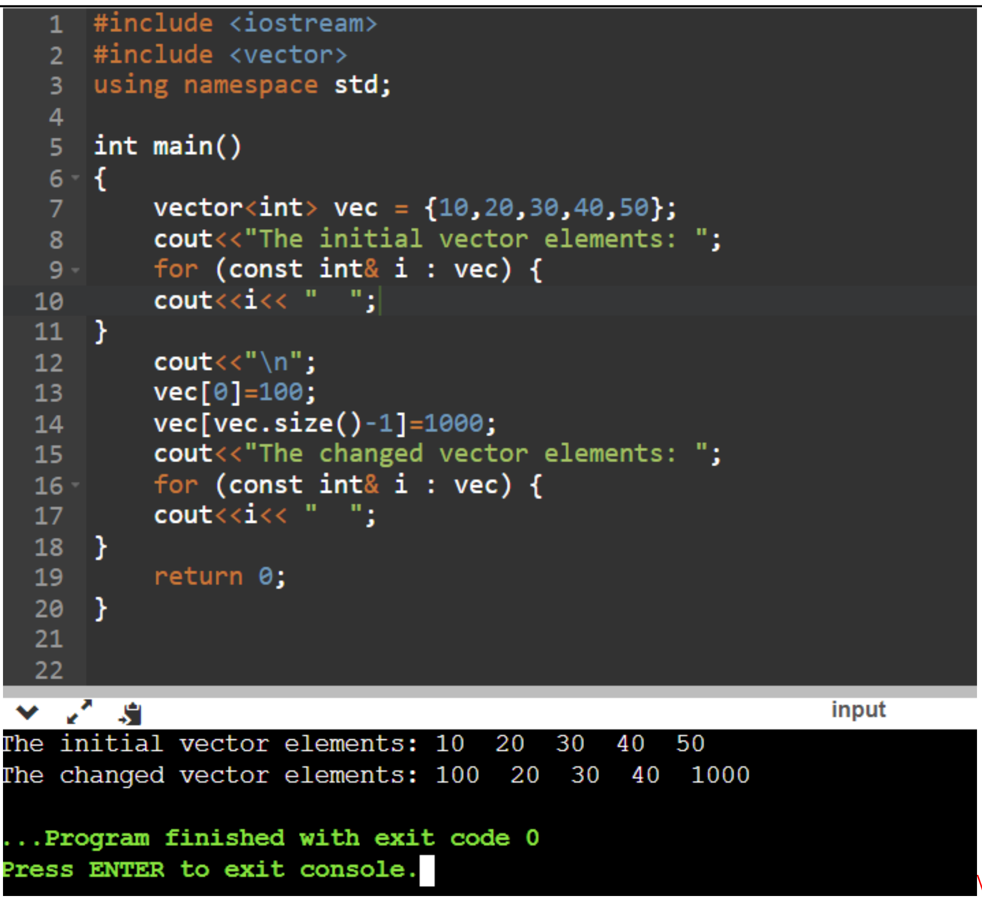
```
#include <iostream>
using namespace std;

int main()
{
    int arr[10] = {};
    cout<<"The initial array elements: ";
    for (int i = 0; i < 10; i++)
        cout<<arr[i]<<"    "; //check before assigning
    cout<<"\n";
    arr[0] = 100; arr[9] = 1000;
    cout<<"The changed array elements: ";
    for (int i = 0; i < 10; i++)
        cout<<arr[i]<<"    "; //check after assigning
    return 0;
}
```

```
The initial array elements: 0    0    0    0    0    0    0    0    0    0
The changed array elements: 100    0    0    0    0    0    0    0    0    1000

...Program finished with exit code 0
Press ENTER to exit console.
```

4. Declaring, Initializing and Accessing an array

Declare a vector of integers named vec and initialize the vector to 10,20,30,40 and 50 Then set the first element of the vector to 100 and the last element of the vector to 1000.

```
#include <iostream>
#include <vector>
using namespace std;

int main()
{
    vector<int> vec = {10,20,30,40,50};
    cout<<"The initial vector elements: ";
    for (const int& i : vec) {
    cout<<i<< "  ";
}
    cout<<"\n";
    vec[0]=100;
    vec[vec.size()-1]=1000;
    cout<<"The changed vector elements: ";
    for (const int& i : vec) {
    cout<<i<< "  ";
}
    return 0;
}
```



```
#include <iostream>
#include <vector>
using namespace std;

int main()
{
    vector<int> vec = {10,20,30,40,50};
    cout<<"The initial vector elements: ";
    for (const int& i : vec) {
    cout<<i<< "  ";
}
    cout<<"\n";
    vec[0]=100;
    vec[vec.size()-1]=1000;
    cout<<"The changed vector elements: ";
    for (const int& i : vec) {
    cout<<i<< "  ";
}
    return 0;
}
```

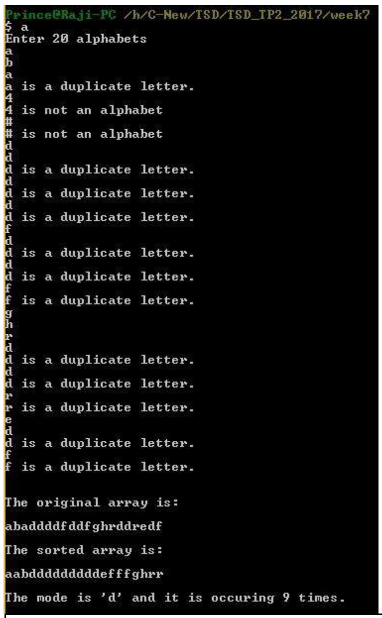
```
The initial vector elements: 10  20  30  40  50
The changed vector elements: 100  20  30  40  1000

...Program finished with exit code 0
Press ENTER to exit console.
```

Name: _______Hong Hai Dang Nguyen_________ Student ID:___103503191____

5. Write a program of the following problem

Create a one-dimensional array to read 20 alphabetical letters (your program should be able to detect and print out an error message if a non-alphabetical letter is entered). As each letter is entered, print a message saying ‘duplicate letter’ if the letter is already in the array. Write a function that can sort the array after all 20 letters have been entered. Write another function that print out the most frequent letter and number of times it was entered. Prepare for the case where all 20 letters are different, or all are the same.

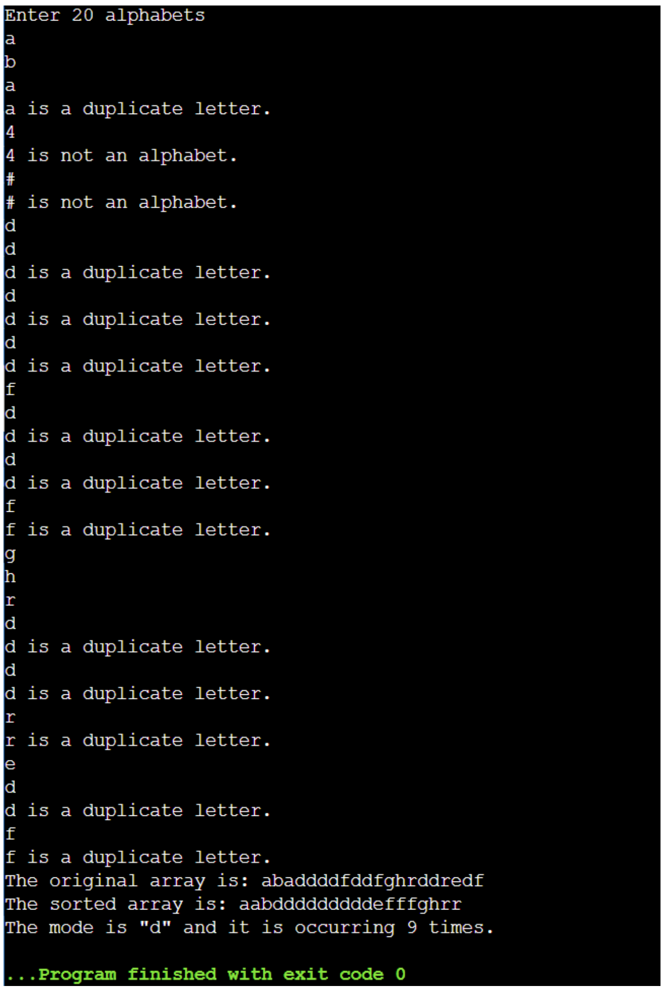
Sample Output:



```
Prince@Raji-PC /h/C-New/TSD/TSD_TP2_2017/week7
$ a
Enter 20 alphabets
a
b
a
a is a duplicate letter.
4
4 is not an alphabet
#
# is not an alphabet
d
d
d is a duplicate letter.
d
d is a duplicate letter.
d
d is a duplicate letter.
f
d
d is a duplicate letter.
d
d is a duplicate letter.
f
f is a duplicate letter.
g
h
r
d
d is a duplicate letter.
d
d is a duplicate letter.
r
r is a duplicate letter.
e
d
d is a duplicate letter.
f
f is a duplicate letter.


The original array is:

abaddddfddfghrddredf

The sorted array is:

aabdddddddddefffghrr

The mode is 'd' and it is occuring 9 times.
```

```
#include <iostream>
using namespace std;

void sort(char arr[], int n)
{
    int i, j;
    char tmp, cmp1, cmp2;
    for(i = 0; i < n; i++)
    {
        for(j = i + 1; j < n; j++)
        {
            cmp1 = arr[i];
            cmp2 = arr[j];

            if(cmp1 >= 'A' && cmp1 <= 'Z')
            {
                //converting temporarily to lower case
                cmp1 = cmp1 + ('a' - 'A');
            }
            if(cmp2 >= 'A' && cmp2 <= 'Z')
            {
                //converting temporarily to lower case
```

```
                    cmp2 = cmp2 + ('a' - 'A');
                }

                if(cmp1 > cmp2) // out of order, swap
                {
                    tmp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = tmp;
                }
            }
        }
    }

int main()
{
    char arr[20];
    int i;

    cout << "Enter 20 alphabets \n";

    i = 0;
    while(i < 20)
    {
        cin >> arr[i];
        if(arr[i] >= 'A' && arr[i] <= 'Z' || arr[i] >= 'a' && arr[i] <= 'z')
        {
            //valid character, check for duplicate
            char t1 = arr[i];
            //converting to lowercase for case insensitive search
            if(t1 >= 'A' && t1 <= 'Z')
            {
                t1 = t1 + ('a' - 'A');
            }
            char t2;
            int j;
            for(j = 0; j < i; j++)
            {
                t2 = arr[j];
                if(t2 >= 'A' && t1 <= 'Z')
                {
                    t2 = t2 + ('a' - 'A');
                }

                if(t1 == t2)
                {
                    cout << arr[i] << " is a duplicate letter. \n";
                    break;
                }
            }

            i++;
        }
        else
        {
            cout << arr[i] << " is not an alphabet. \n";
        }
    }

    cout << "The original array is: ";
    for(i = 0; i < 20; i++)
        cout << arr[i];

    cout<<"\n";
    sort(arr, 20);

    cout << "The sorted array is: ";
    for(i = 0; i < 20; i++)
        cout << arr[i];

    int freq[26]; // to store frequency of all letters
    for(i = 0; i < 26; i++)
        freq[i] = 0;

    for(i = 0; i < 20; i++)
    {
        char c = arr[i];
        //convert to lower case
        if(c >= 'A' && c <= 'Z')
            c = c + ('a' - 'A');
        //update count
        freq[c - 'a'] ++;
    }

    int max = 0;
    for(i = 0; i < 26; i++)
    {
        if(freq[i] > max)
            max = freq[i];
    }

    //are there multiple max values?
    int maxcount = 0;
    for(i = 0; i < 26; i++)
    {
        if(freq[i] == max)
            maxcount ++;
    }

    cout << "\n";
    if(maxcount == 1)
    {
        cout << "The mode is \"";
        for(i = 0; i < 26; i++)
        {
            if(freq[i] == max)
            {
                cout << (char)('a' + i);
                break;
            }
        }
        cout << "\" and it is occurring " << max << " times.";
    }
    else
    {
        cout << "The modes are \"";
        for(i = 0; i < 26; i++)
        {
            if(freq[i] == max)
            {
                cout << (char)('a' + i);
                if(maxcount > 1)
                {
                    cout << ", ";
                    maxcount --;
                }
            }
        }
        cout << "\" and they occur " << max << " times";
    }
    return 0;
}
```



```
Enter 20 alphabets
a
b
a
a is a duplicate letter.
4
4 is not an alphabet.
#
# is not an alphabet.
d
d
d is a duplicate letter.
d
d is a duplicate letter.
d
d is a duplicate letter.
f
d
d is a duplicate letter.
d
d is a duplicate letter.
f
f is a duplicate letter.
g
h
r
d
d is a duplicate letter.
d
d is a duplicate letter.
r
r is a duplicate letter.
e
d
d is a duplicate letter.
f
f is a duplicate letter.
The original array is: abaddddfddfghrddredf
The sorted array is: aabddddddddefffghrr
The mode is "d" and it is occurring 9 times.

...Program finished with exit code 0
```

Name: _______Hong Hai Dang Nguyen_________ Student ID:___103503191____

Report (SWE20004)

Write a one-page report on this lab covering the following:

1. Summarize the topics you explored and the activities you did during this lab.
2. Classify (group) these topics and actions under appropriate headings. Do not just copy the headings used in the instructions. For example, explain the following, what are the following commands do?
3. Discuss the relevance of these topics and actions in terms of C++ programming. i.e. How do the things in this lab work contribute to your understanding of C++ programming overall?
4. Why do you need to understand (and use)Arrays and vectors ?

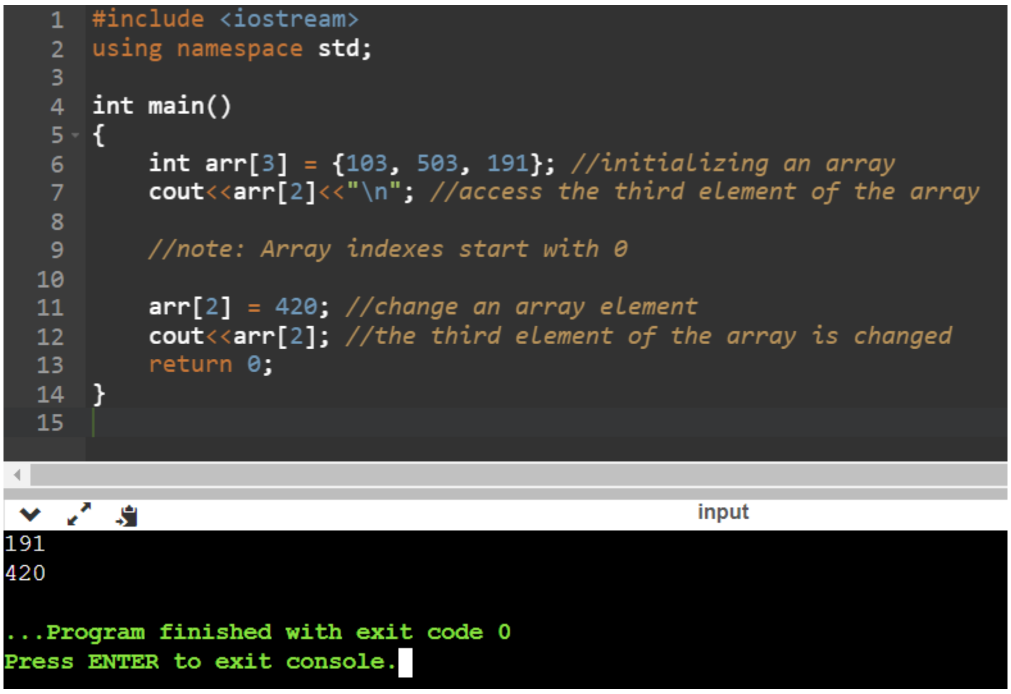
This report is worth 5% towards your unit assessment. Use the below page as a template. Either you can type it or write it in your words.

Introduction

This week's report will illustrate the topics covered and the activities of the lab. More specifically, we were taught about the use of arrays and vectors. In addition, we went over some of the exercises in this week's lab in order to strengthen our knowledge and understanding of the topics.

Arrays

Arrays are used to hold many values in a single variable instead of creating distinct variables for each value. An array can be declared by indicating the array's name in square brackets, the type of variable to be used, and the number of elements it should hold. To insert values, enclose the values in curly brackets enclosed by commas. The following example program will indicate how to initialize an array, access an element of an array and change an array element:



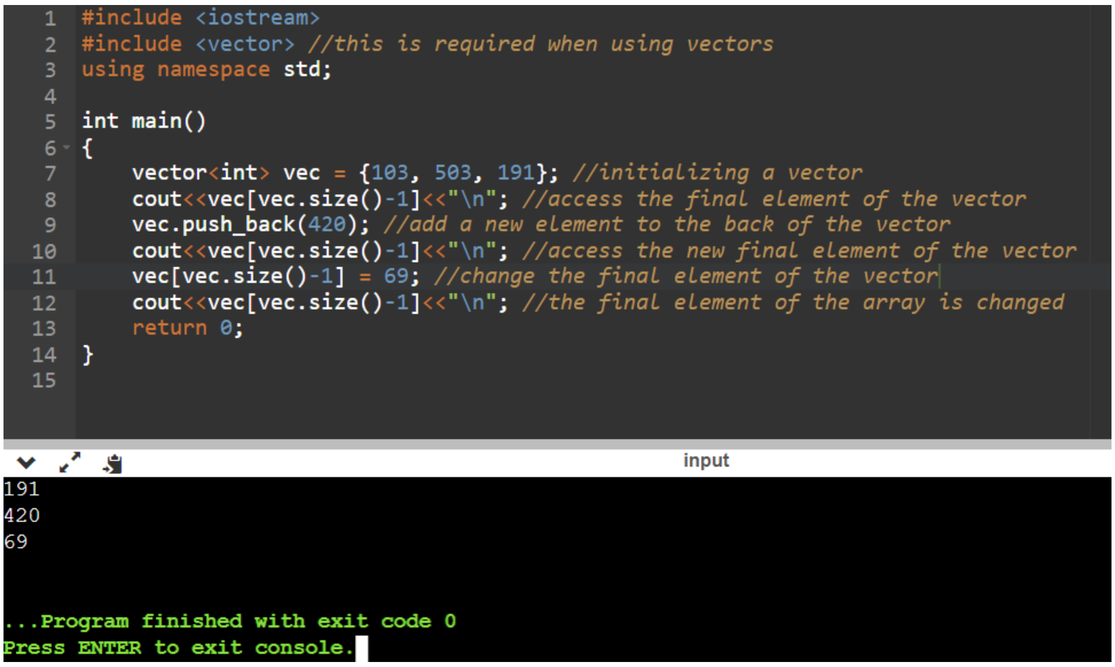
```cpp
#include <iostream>
using namespace std;

int main()
{
    int arr[3] = {103, 503, 191}; //initializing an array
    cout<<arr[2]<<"\n"; //access the third element of the array

    //note: Array indexes start with 0

    arr[2] = 420; //change an array element
    cout<<arr[2]; //the third element of the array is changed
    return 0;
}
```

input

```
191
420

...Program finished with exit code 0
Press ENTER to exit console.
```

Vectors

Vectors are the dynamic arrays used to store values. In contrast to arrays, which store sequential data and are static, vectors give the programmers greater flexibility. Vectors can automatically alter their size when an element is added or removed. The elements of a vector are placed in contiguous storage so they can be retrieved and traversed using iterators. The following example program will show how to initialize a vector, access an element of the vector, add a value to a vector and change the value of the vector:



```cpp
#include <iostream>
#include <vector> //this is required when using vectors
using namespace std;

int main()
{
    vector<int> vec = {103, 503, 191}; //initializing a vector
    cout<<vec[vec.size()-1]<<"\n"; //access the final element of the vector
    vec.push_back(420); //add a new element to the back of the vector
    cout<<vec[vec.size()-1]<<"\n"; //access the new final element of the vector
    vec[vec.size()-1] = 69; //change the final element of the vector
    cout<<vec[vec.size()-1]<<"\n"; //the final element of the array is changed
    return 0;
}
```

input

```
191
420
69

...Program finished with exit code 0
Press ENTER to exit console.
```

Lab activities

This week's lab exercises were significantly challenging, as arrays and vectors are difficult and confusing to use. Especially exercise 5, which I consider it to be the hardest problem so far in this unit and it took me a large amount of time to finish it. Exercise 2 is also difficult to solve, as to identify the errors we need to pay attention to details, such as the right use of the semicolon.

The importance of this week’s topic

Knowing how to effectively and efficiently use arrays and vectors in programming will help us to store values (that are the same in essence) instead of assigning values to different variables, which will help us to save time and to be able to access to the values when needed.

Conclusion

Overall, this week's lab helped us to understand the use of arrays and vectors. I could now confidently use arrays or vectors when I need to store values and use it when necessary, which will save my time and effort.