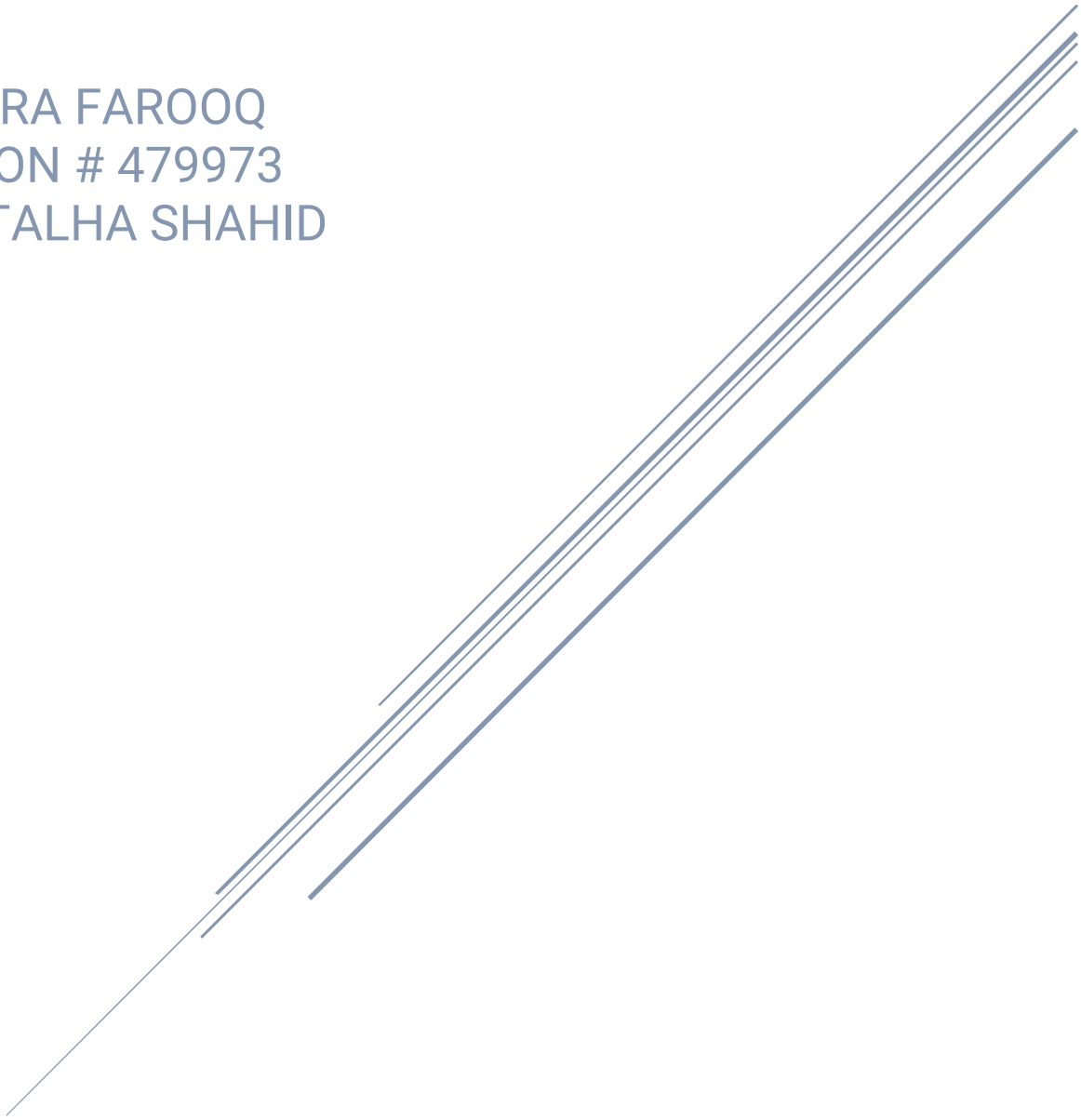
FUNDAMENTALS OF PROGRAMING

ASSIGNMENT:01

NAME:BUSHRA FAROOQ
REGISTRATION # 479973
ACCESSOR:TALHA SHAHID

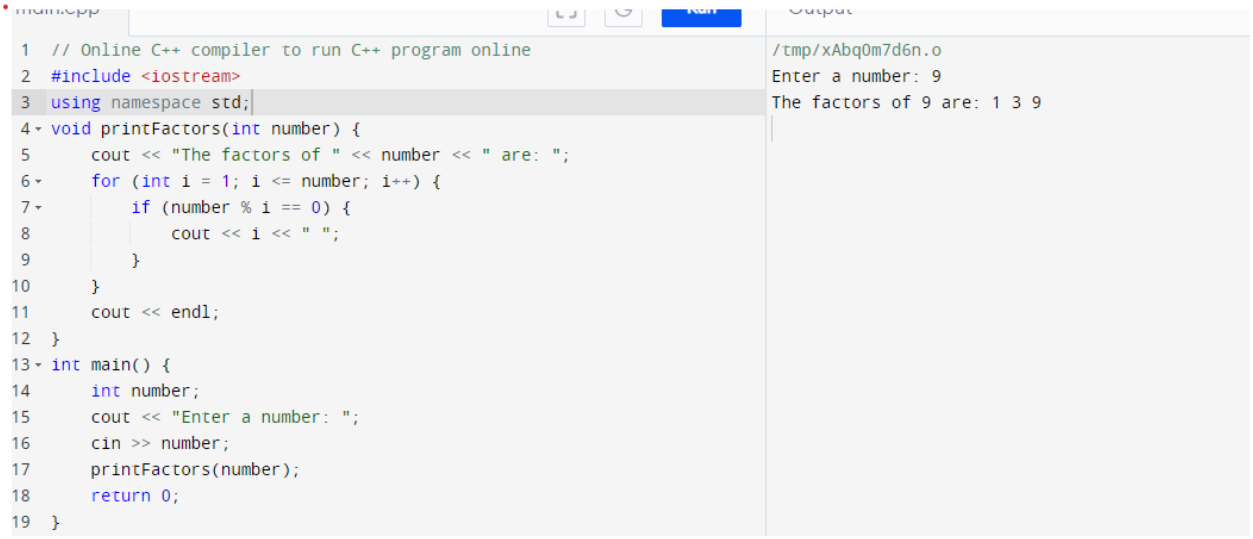


QUESTION # 01

Write a C++ program to display factors of a number using for loops.

ANSWER:

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

void printFactors(int number) {
    cout << "The factors of " << number << " are: ";
    for (int i = 1; i <= number; i++) {
        if (number % i == 0) {
            cout << i << " ";
        }
    }
    cout << endl;
}
int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
  printFactors(number);
    return 0;
}
```



```cpp
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;
void printFactors(int number) {
    cout << "The factors of " << number << " are: ";
    for (int i = 1; i <= number; i++) {
        if (number % i == 0) {
            cout << i << " ";
        }
    }
    cout << endl;
}
int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
    printFactors(number);
    return 0;
}
```

```
/tmp/xAbq0m7d6n.o
Enter a number: 9
The factors of 9 are: 1 3 9
```

QUESTION#02

Write output to the following code.

```
#include <iostream>
int main() {
 int x = 5;
 int y = 10;
 if (x == 5)
 if (y == 10)
 std::cout << "x is 5 and y is 10" << std::endl;
 else
 std::cout << "x is not 5" << std::endl;
 return 0;
}
```

ANSWER:

QUESTION#03

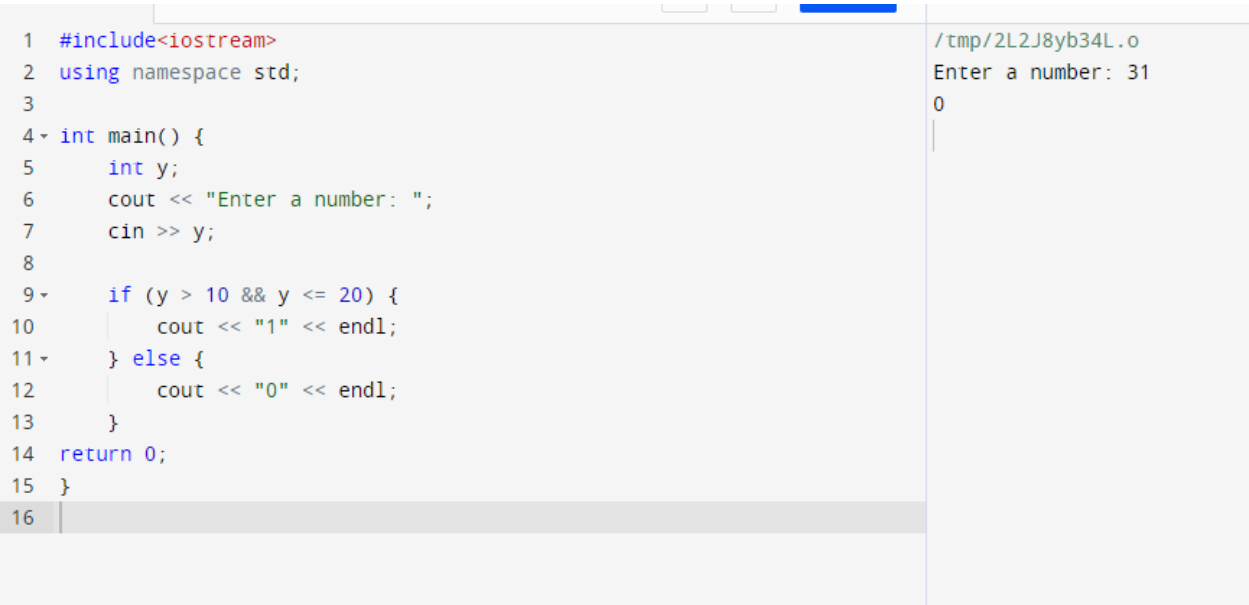
Write a C++ program, take an integer value from user and check if it's greater than 10 and less thanequal to 20. Print 1 if yes and print 0 if no. Use appropriate datatype for output.

ANSWER:

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

int main() {
   int y;
   cout << "Enter a number: ";
   cin >> y;

   if (y > 10 && y <= 20) {
      cout << "1" << endl;
   } else {
      cout << "0" << endl;
   }
```

return 0;

}



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

int main() {
    int y;
    cout << "Enter a number: ";
    cin >> y;

    if (y > 10 && y <= 20) {
        cout << "1" << endl;
    } else {
        cout << "0" << endl;
    }
return 0;
}
```

```
/tmp/2L2J8yb34L.o
Enter a number: 31
0
```

QUESTION # 04

Write a C++ program that uses a while loop to find the largest prime number less than a given positive integer N. Your program should take the value of N as input from the user and then find the largest prime number less than or equal to N. You are not allowed to use any library or pre-existing functions to check for prime numbers.

ANSWER:

#include<iostream>

using namespace std;

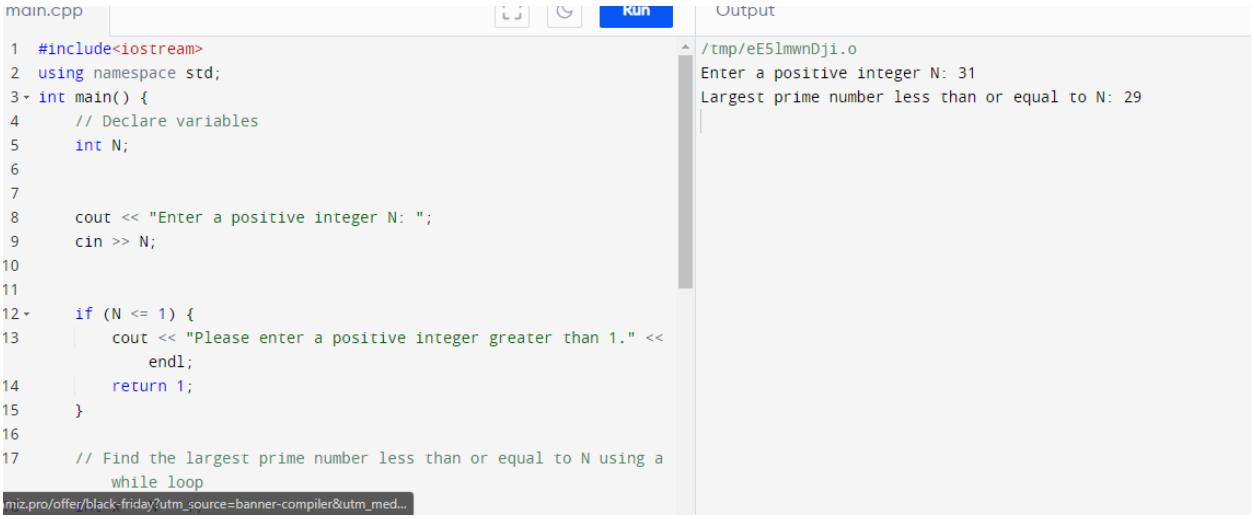
int main() {

// Declare variables

int N;

```
cout << "Enter a positive integer N: ";
cin >> N;



if (N <= 1) {
   cout << "Please enter a positive integer greater than 1." << endl;
   return 1;
}

// Find the largest prime number less than or equal to N using a while loop
int x = N - 1;
while (x > 1) {

   bool isPrime = true;
   for (int i = 2; i * i <= x; ++i) {
      if (x % i == 0) {
         isPrime = false;
         break;
      }
   }

   // If the x is prime, print it and exit the loop
   if (isPrime) {
      cout << "Largest prime number less than or equal to N: " << x << endl;
      break;
   }

   // Move to the next number
   --x;
```

```
    }

    return 0;
}
```



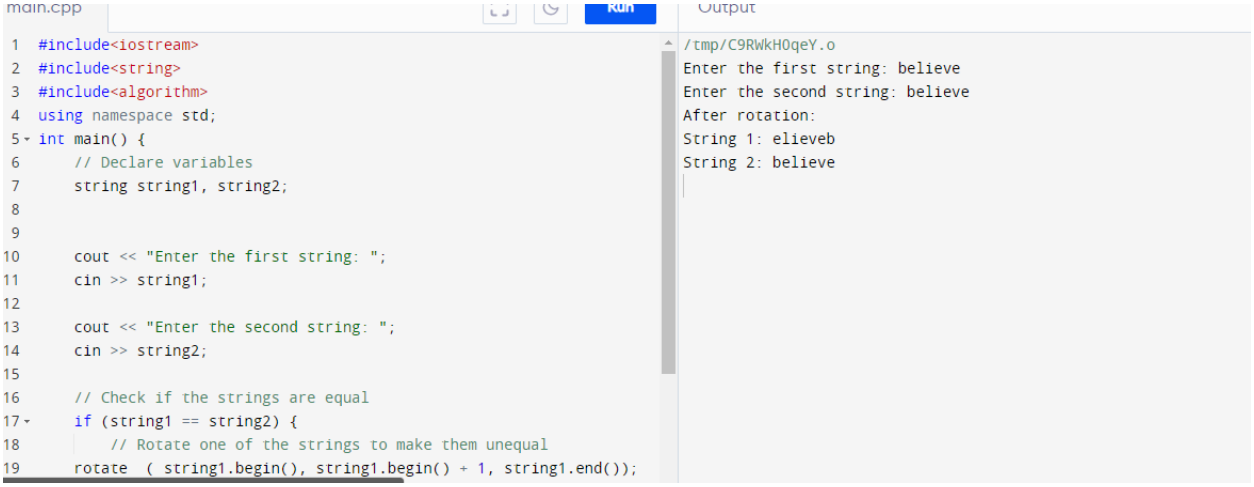
QUESTION #05

Write a C++ program, take two string as input from user and check if both strings are equal or not. If they are equal make them unequal by rotating string. e.g., Hello is turned into olleH etc.

ANSWER:

```
#include<iostream>
#include<string>
#include<algorithm>
using namespace std;
int main() {
    // Declare variables
    string string1, string2;
```

```
    cout << "Enter the first string: ";
    cin >> string1;

    cout << "Enter the second string: ";
    cin >> string2;

    // Check if the strings are equal
    if (string1 == string2) {
        // Rotate one of the strings to make them unequal
    rotate  ( string1.begin(), string1.begin() + 1, string1.end());

        // Print the rotated strings
        cout << "After rotation:\n";
        cout << "String 1: " << string1 << endl;
        cout << "String 2: " << string2 << endl;
    } else {
        cout << "The strings are already unequal." << endl;
    }

    return 0;
}
```



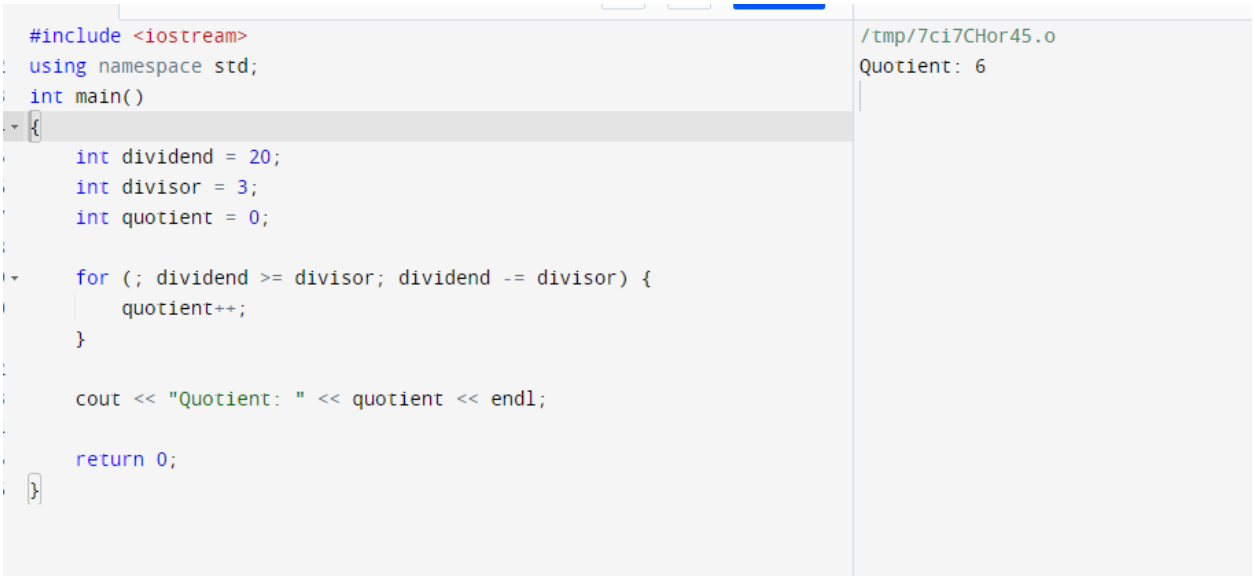
QUESTION # 06

Perform division in C++ without / using for loops. You can use / only to display the final results. Your dividend must be greater than divisor.

ANSWER:

```
#include <iostream>
using namespace std;
int main()
{
   int dividend = 20;
   int divisor = 3;
   int quotient = 0;

   for (; dividend >= divisor; dividend -= divisor) {
      quotient++;
   }

   cout << "Quotient: " << quotient << endl;

   return 0;
```

}



```
#include <iostream>
using namespace std;
int main()
{
    int dividend = 20;
    int divisor = 3;
    int quotient = 0;

    for (; dividend >= divisor; dividend -= divisor) {
        quotient++;
    }

    cout << "Quotient: " << quotient << endl;

    return 0;
}
```

```
/tmp/7ci7CHor45.o
Quotient: 6
```

QUESTION # 07

Write a C++program for a string which may contain lowercase and uppercase characters. The task is to remove all duplicate characters from the string and find the resultant string.

ANSWER:

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

int main() {
    string abc;
    cout << "Enter a string: ";
cin >> abc;

    unordered_set<char> seenChars;
    string result;
```

```
    for (char ch : abc) {
        if (seenChars.insert(ch).second) {
            result += ch;
        }
    }

    cout << "Resultant string after removing duplicates: " << result << endl;

    return 0;
}
```

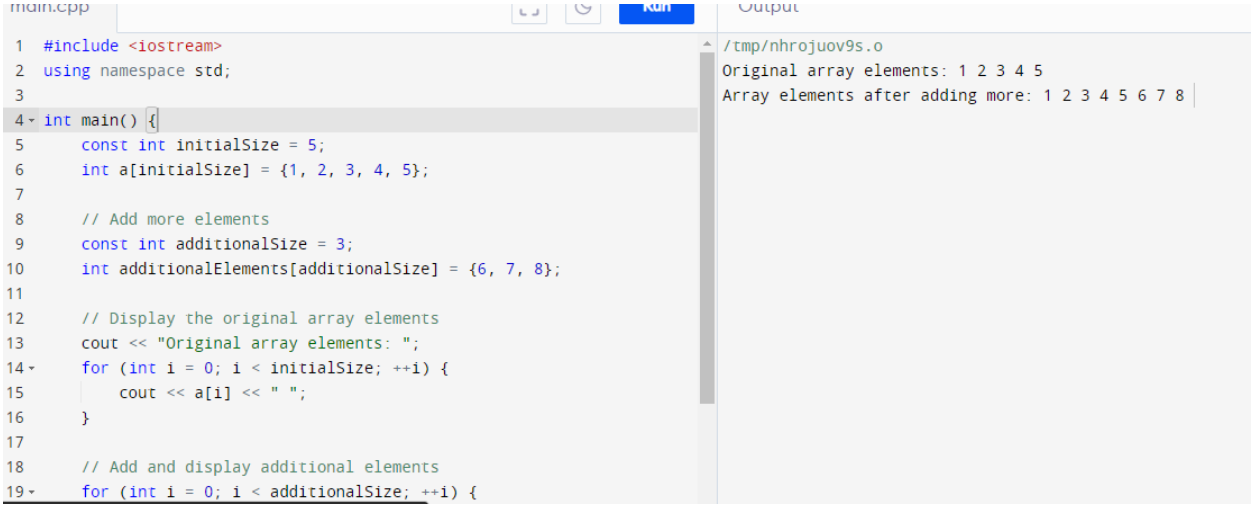
QUESTION # 08

Suppose an integer array a[5] = {1,2,3,4,5}. Add more elements to it and display them in C++.

ANSWER:

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

```
int main() {
    const int initialSize = 5;
    int a[initialSize] = {1, 2, 3, 4, 5};

    // Add more elements
    const int additionalSize = 3;
    int additionalElements[additionalSize] = {6, 7, 8};

    // Display the original array elements
    cout << "Original array elements: ";
    for (int i = 0; i < initialSize; ++i) {
        cout << a[i] << " ";
    }

    // Add and display additional elements
    for (int i = 0; i < additionalSize; ++i) {
        a[initialSize + i] = additionalElements[i];
    }

    cout << "\nArray elements after adding more: ";
    for (int i = 0; i < initialSize + additionalSize; ++i) {
        cout << a[i] << " ";
    }

    return 0;
}
```



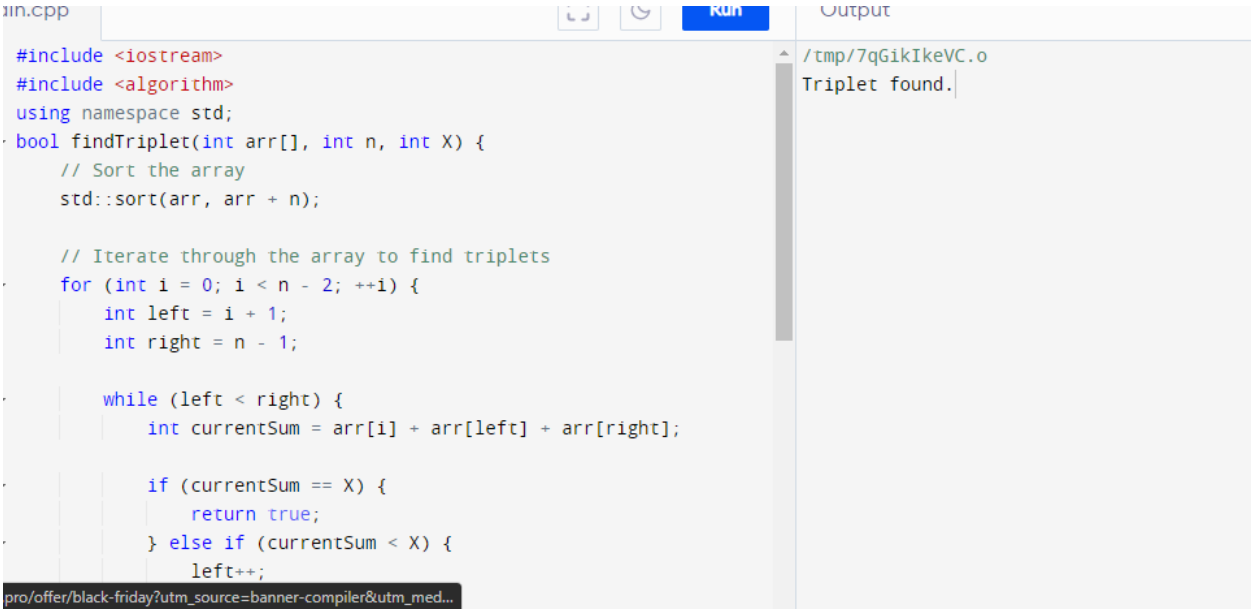
QUESTION # 09

Given an integer array and an integer X. Find if there's a triplet in the array which sums up to the given integer X.

ANSWER:

```
#include <iostream>
#include <algorithm>
using namespace std;
bool findTriplet(int arr[], int n, int X) {
    // Sort the array
    std::sort(arr, arr + n);

    // Iterate through the array to find triplets
    for (int i = 0; i < n - 2; ++i) {
        int left = i + 1;
        int right = n - 1;

        while (left < right) {
            int currentSum = arr[i] + arr[left] + arr[right];
```

```
            if (currentSum == X) {
                return true;
            } else if (currentSum < X) {
                left++;
            } else {
                right--;
            }
        }
    }

    return false;
}

int main() {
    // Example array
    int myArray[] = {2, 4, 67, 6, 9, 8};
    int arraySize = sizeof(myArray) / sizeof(myArray[0]);
    int targetSum = 23;

    // Check if a triplet with the given sum exists
    if (findTriplet(myArray, arraySize, targetSum)) {
        cout << "Triplet found.";
    } else {
        cout << "Triplet not found.";
    }

    Return 0;
}
```



QUESTION # 10

Implement Bubble Sort on an array of 6 integers.

ANSWER:

```
#include <iostream>
using namespace std;
void bubbleSort(int arr[], int n) {
    for (int a = 0; a < n - 1; ++a) {
        for (int k = 0; k < n - a - 1; ++k) {
            if (arr[k] > arr[k + 1]) {
                // Swap elements if they are in the wrong order
                int temp = arr[k];
                arr[k] = arr[k + 1];
                arr[k + 1] = temp;
```

```
            }
        }
    }
}

int main() {
    // Example array
    int myArray[] = {6, 2, 4, 1, 5, 3};
    int arraySize = sizeof(myArray) / sizeof(myArray[0]);

    // Applying Bubble Sort
    bubbleSort(myArray, arraySize);

    // Displaying the sorted array
    cout << "Sorted array: ";
    for (int a = 0; a < arraySize; ++a) {
        cout << myArray[a] << " ";
    }

    return 0;
}
```

main.cpp

Run

Output

```cpp
#include <iostream>
using namespace std;
void bubbleSort(int arr[], int n) {
    for (int a = 0; a < n - 1; ++a) {
        for (int k = 0; k < n - a - 1; ++k) {
            if (arr[k] > arr[k + 1]) {
                // Swap elements if they are in the wrong order
                int temp = arr[k];
                arr[k] = arr[k + 1];
                arr[k + 1] = temp;
            }
        }
    }
}

int main() {
    // Example array
```

```
/tmp/AAi7K4YEcW.o
Sorted array: 1 2 3 4 5 6
```