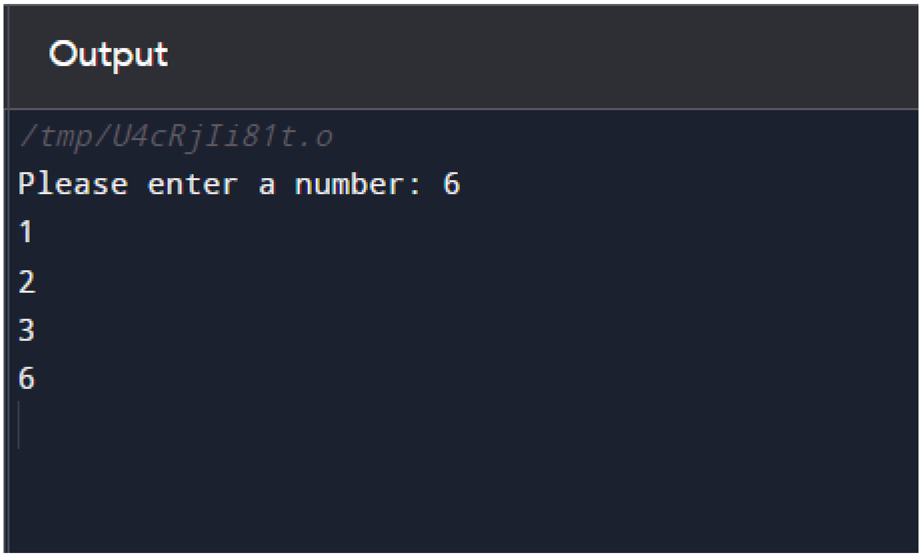
1.Write a C++ program to display factors of a number using for loops.

```
#include <iostream>
using namespace std;
int main()
{
        int n;
        cout << "Please enter a number: ";
        cin >> n;
        for(int i=1; i<=n; i++)
        {
           if(n % i ==0 )
           cout<<i;
           else
           continue;
           cout<<endl;
        }
        return 0;
}
```



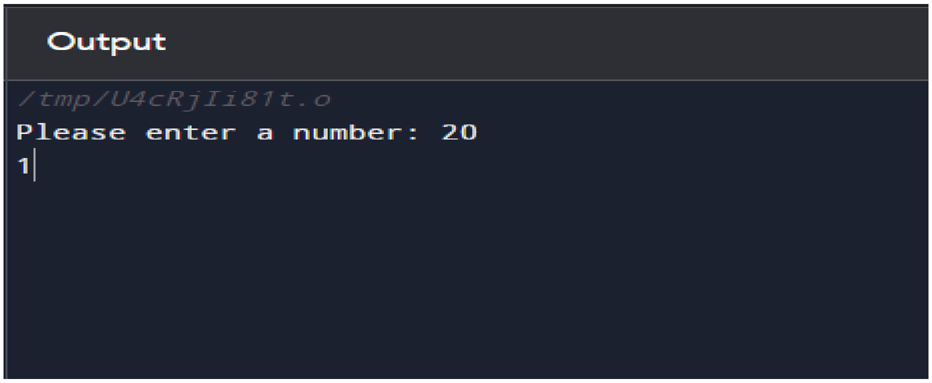
```
/tmp/U4cRjIi81t.o
Please enter a number: 6
1
2
3
6
```

2.Write output to the following code.

x is 5 and y is 10

3.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.

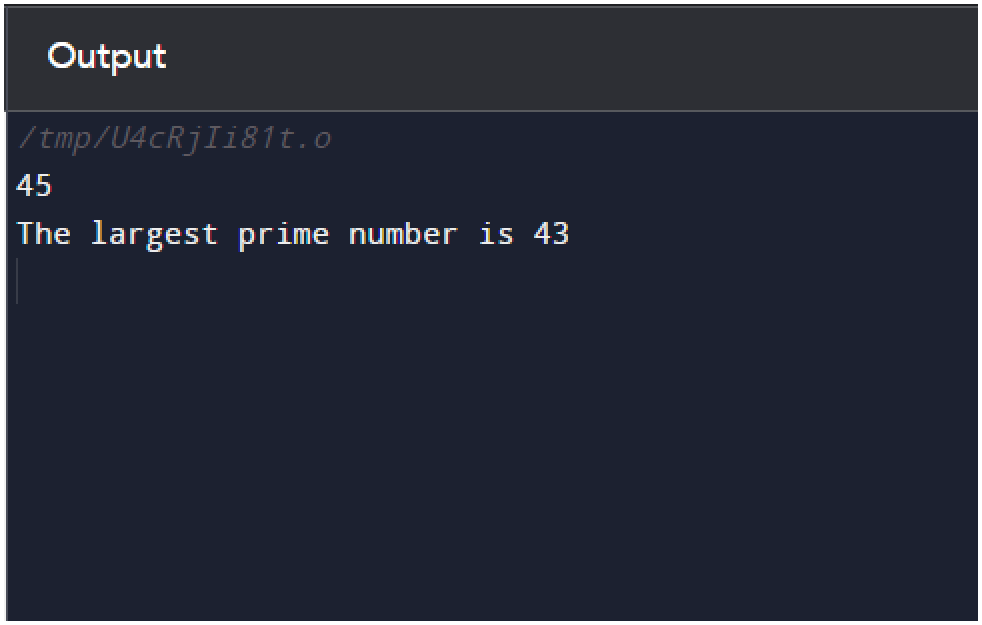
```
#include <iostream>

using namespace std;

int main()

{

        int n;

        cout << "Please enter a number: ";

        cin >> n;


        if(n>10 && n<=20)

        cout << "1";

        else

        cout << "0";


        return 0;

}
```



```
/tmp/U4cRjIi81t.o
Please enter a number: 20
1
```

4.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.

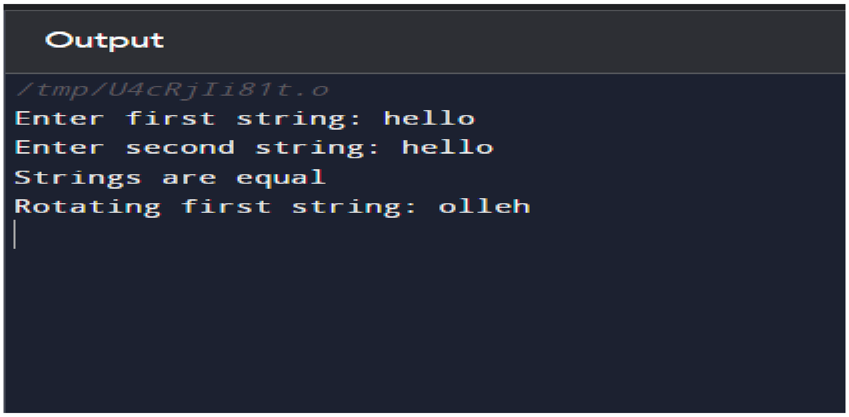
```
#include <iostream>
using namespace std;
int main()
{
    int n = 0;
    cin >> n;
    int count = 2;
    int i = 1;
    int largestPrime = 1;
    bool isPrime = true;
    while (count <= n)
    {
        isPrime = true;
        i = 2;
        while (i < count && isPrime == true)
        {
            if (!(count % i == 0))
            {
                isPrime = true;
            }
            else
            {
                isPrime = false;
            }
            i++;
        }
        if (isPrime == true)
        {
            largestPrime = count;
        }
        count++;
    }
    cout << "The largest prime number is " << largestPrime << endl;
    return 0;
}
```



```
/tmp/U4cRjIi81t.o
45
The largest prime number is 43
```

5.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.

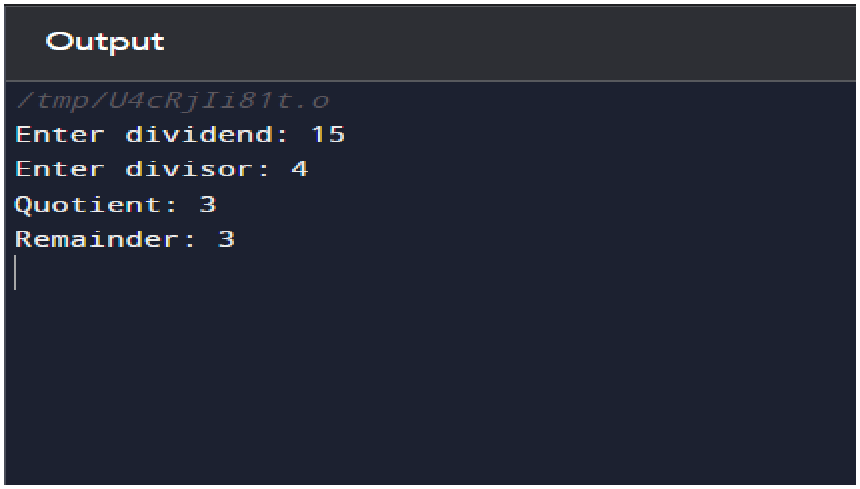
```
#include <iostream>

#include <algorithm>

using namespace std;

int main() {

  string str1, str2;

  cout << "Enter first string: ";

  cin >> str1;

  cout << "Enter second string: ";

  cin >> str2;

  if(str1 == str2) {

    cout << "Strings are equal";

    cout<<endl;

    reverse(str1.begin(), str1.end());

    cout << "Rotating first string: " << str1 << endl;

  } else {

    cout << "Strings are not equal";

    cout<<endl;

  }

  return 0;}
```



```
/tmp/U4cRjIi81t.o
Enter first string: hello
Enter second string: hello
Strings are equal
Rotating first string: olleh
```

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

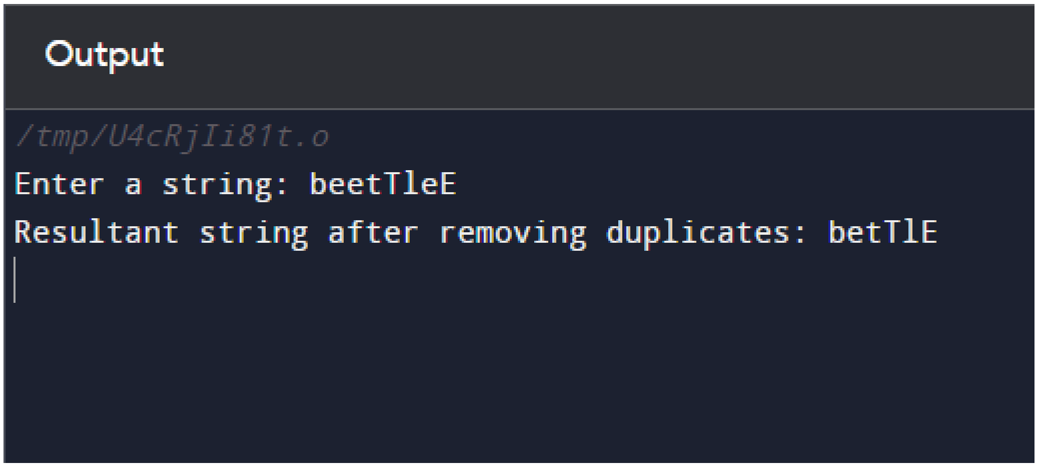
```
#include <iostream>
using namespace std;
int main() {
   int dividend, divisor;
   cout << "Enter dividend: ";
   cin >> dividend;
   cout << "Enter divisor: ";
   cin >> divisor;
   int quotient = 0;
   int remainder = dividend;
   while (remainder >= divisor) {
      remainder -= divisor;
      quotient++;
   }
   cout << "Quotient: " << quotient << endl;
   cout << "Remainder: " << remainder << endl;
   return 0;
}
```



```
/tmp/U4cRjIi81t.o
Enter dividend: 15
Enter divisor: 4
Quotient: 3
Remainder: 3
```

7.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.

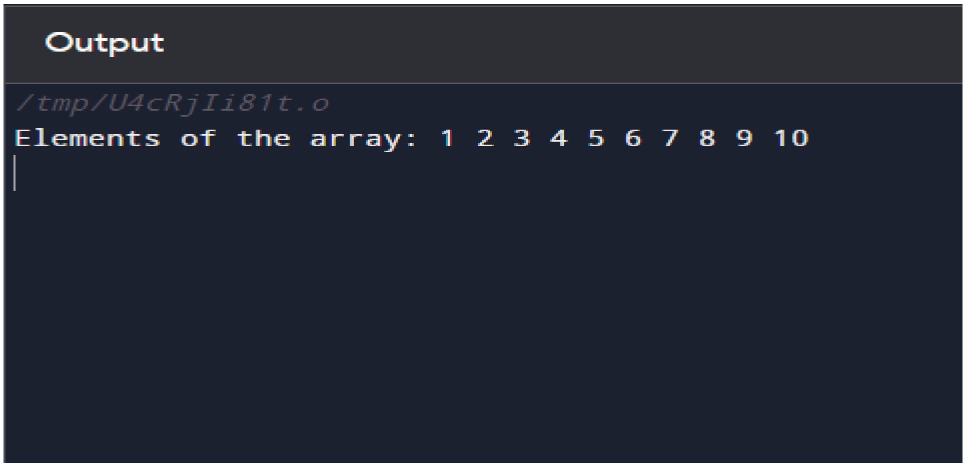
```
#include <iostream>
#include <string>
#include <unordered_set>
using namespace std;
string removeDuplicates(string str) {
    string result = "";
    unordered_set<char> seen;
    for (char ch : str) {
        if (seen.find(ch) == seen.end()) {
            result += ch;
            seen.insert(ch);
        }
    }
    return result;
}
int main() {
    string input;
    cout << "Enter a string: ";
    cin >> input;
    string result = removeDuplicates(input);
    cout << "Resultant string after removing duplicates: " << result << endl;
    return 0;
}
```



```
/tmp/U4cRjIi81t.o
Enter a string: beetTleE
Resultant string after removing duplicates: betTlE
```

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

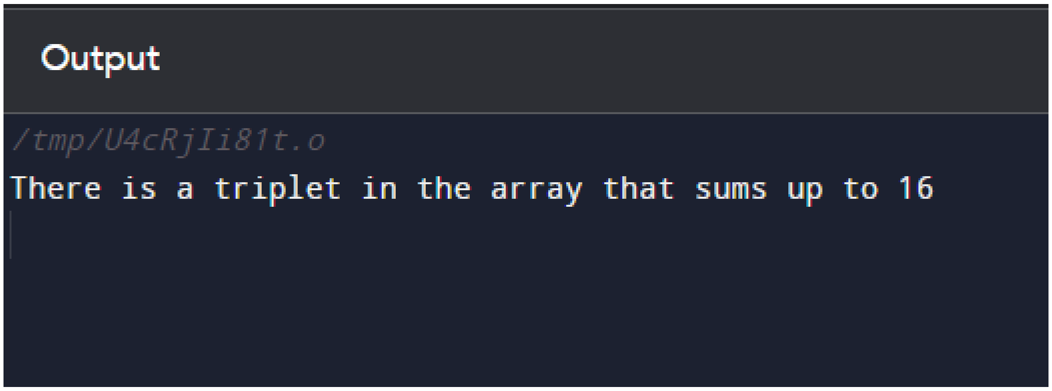
```
#include <iostream>
using namespace std;

int main() {
   int a[10] = {1, 2, 3, 4, 5};
   for (int i = 5; i < 10; i++) {
      a[i] = i + 1;
   }
   cout << "Elements of the array: ";
   for (int i = 0; i < 10; i++) {
      cout << a[i] << " ";
   }
   cout << endl;

   return 0;
}
```



```
/tmp/U4cRjIi81t.o
Elements of the array: 1 2 3 4 5 6 7 8 9 10
```

9.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.

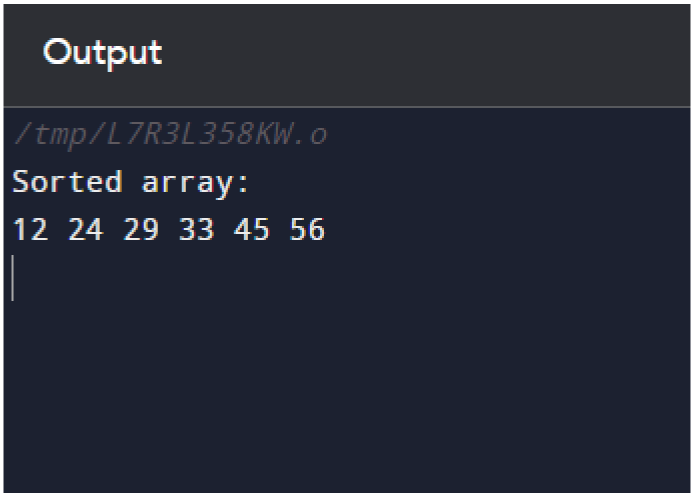
```
#include <iostream>

using namespace std;

bool findTriplet(int arr[], int n, int targetSum) {

    for (int i = 0; i < n - 2; i++) {

        for (int j = i + 1; j < n - 1; j++) {

            for (int k = j + 1; k < n; k++) {

                if (arr[i] + arr[j] + arr[k] == targetSum) {

                    return true;

                }

            }

        }

    }

    return false;

}

int main() {

    int arr[] = {1, 4, 2, 10, 5, 3};

    int targetSum = 16;
    int size = sizeof(arr) / sizeof(arr[0]);


    if (findTriplet(arr, size, targetSum)) {
        cout << "There is a triplet in the array that sums up to " << targetSum << endl;
    } else {
        cout << "There is no triplet in the array that sums up to " << targetSum << endl;
    }


    return 0;
}
```



```
/tmp/U4cRjIi81t.o
There is a triplet in the array that sums up to 16
```

10.Implement Bubble Sort on an array of 6 integers

```
#include <iostream>
using namespace std;
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n-1; 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;
            }
        }
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int main() {
    int arr[] = {24, 29, 12, 56, 33, 45};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    cout<<"Sorted array: \n";
    printArray(arr, n);
    return 0;
}
```



```
/tmp/L7R3L358KW.o
Sorted array:
12 24 29 33 45 56
```