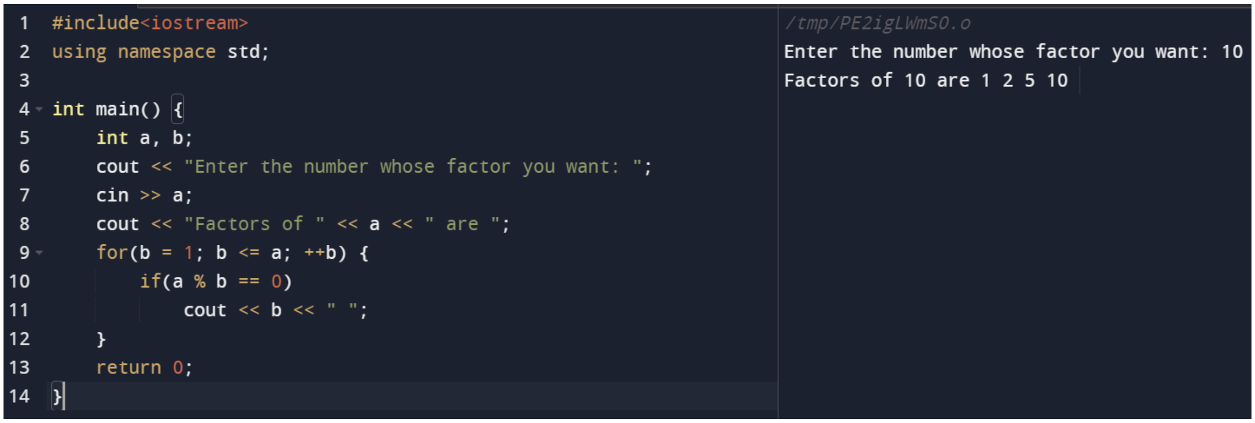
1. Write a C++ program to display factors of a number using for loops.

Code:

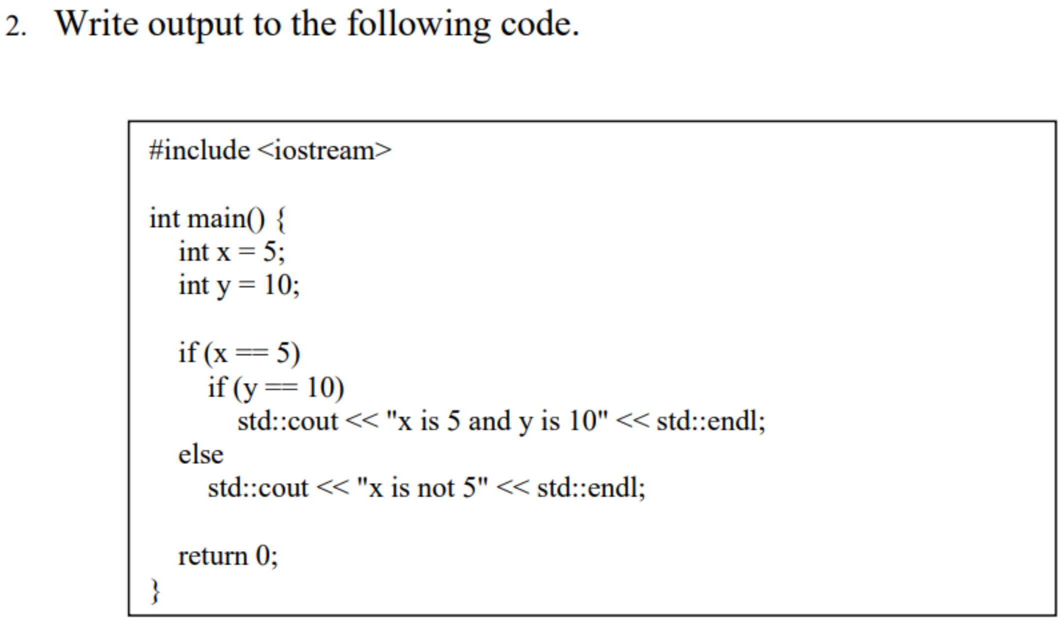
```
#include<iostream>
using namespace std;

int main() {
   int a, b;
   cout << "Enter the number whose factor you want: ";
   cin >> a;
   cout << "Factors of " << a << " are ";
   for(b = 1; b <= a; ++b) {
      if(a % b == 0)
         cout << b << " ";
   }
   return 0;
}
```



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

int main() {
    int a, b;
    cout << "Enter the number whose factor you want: ";
    cin >> a;
    cout << "Factors of " << a << " are ";
    for(b = 1; b <= a; ++b) {
        if(a % b == 0)
            cout << b << " ";
    }
    return 0;
}
```

```
/tmp/PE2igLWmS0.o
Enter the number whose factor you want: 10
Factors of 10 are 1 2 5 10
```

2.



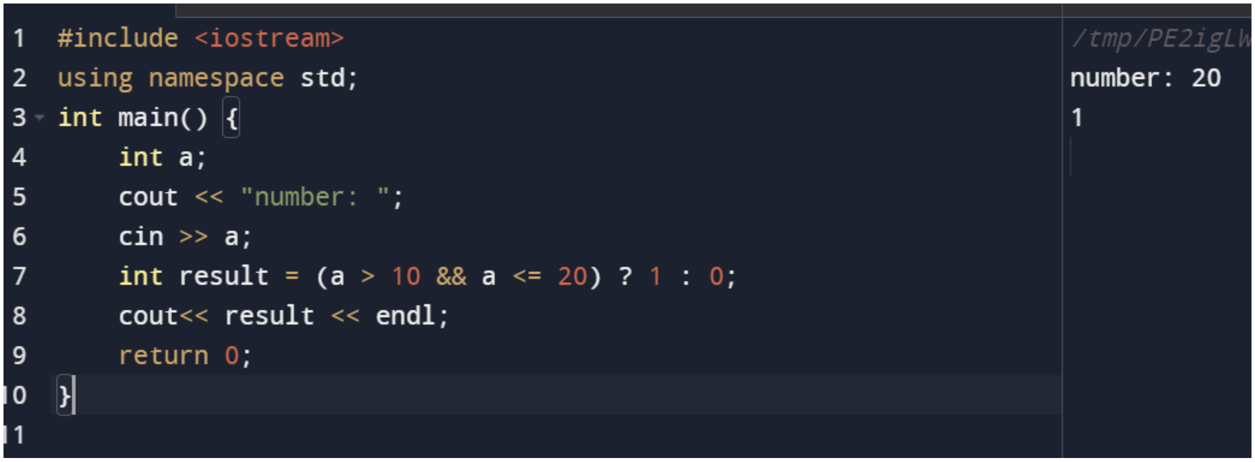
2. Write output to the following code.

```
#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;
}
```

Output of this code is: " 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 than equal to 20. Print 1 if yes and print 0 if no. Use appropriate datatype for output.

```
#include <iostream>
using namespace std;
int main() {
   int a;
   cout << "number: ";
   cin >> a;
   int result = (a > 10 && a <= 20) ? 1 : 0;
   cout<< result << endl;
   return 0;
}
```

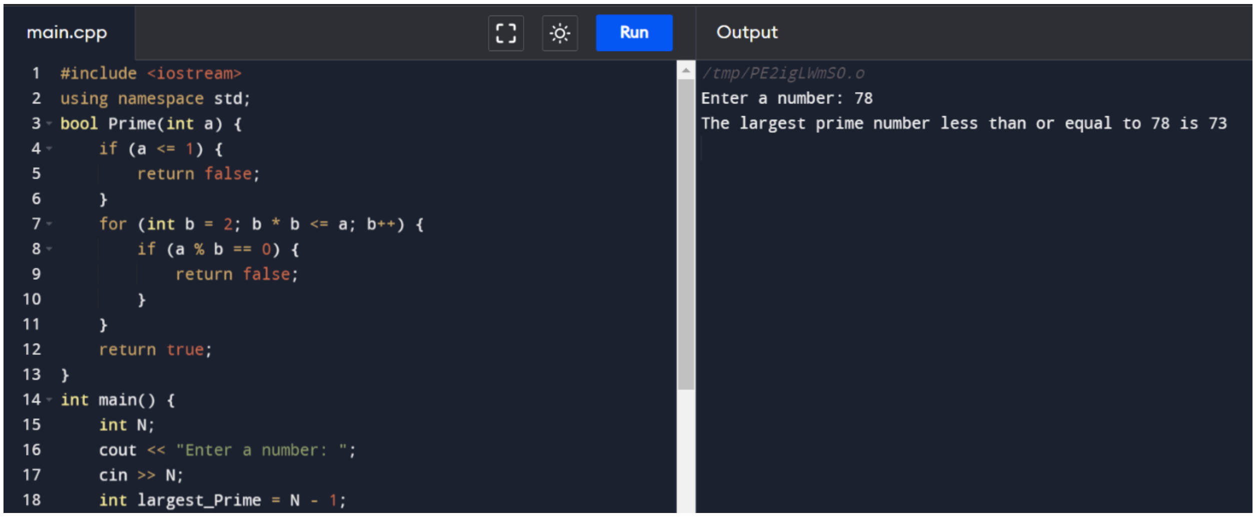


```
#include <iostream>
using namespace std;
int main() {
    int a;
    cout << "number: ";
    cin >> a;
    int result = (a > 10 && a <= 20) ? 1 : 0;
    cout<< result << endl;
    return 0;
}
```

```
/tmp/PE2igLW
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;
bool Prime(int a) {
  if (a <= 1) {
    return false;
  }
  for (int b = 2; b * b <= a; b++) {
    if (a % b == 0) {
      return false;
    }
  }
  return true;
}
int main() {
  int N;
  cout << "Enter a number: ";
  cin >> N;
  int largest_Prime = N - 1;
  while (largest_Prime > 1) {
    if (Prime(largest_Prime)) {
      break;
    }
    largest_Prime--;
  }
  cout << "The largest prime number less than or equal to " << N << " is " << largest_Prime << endl;
  return 0;
}
```

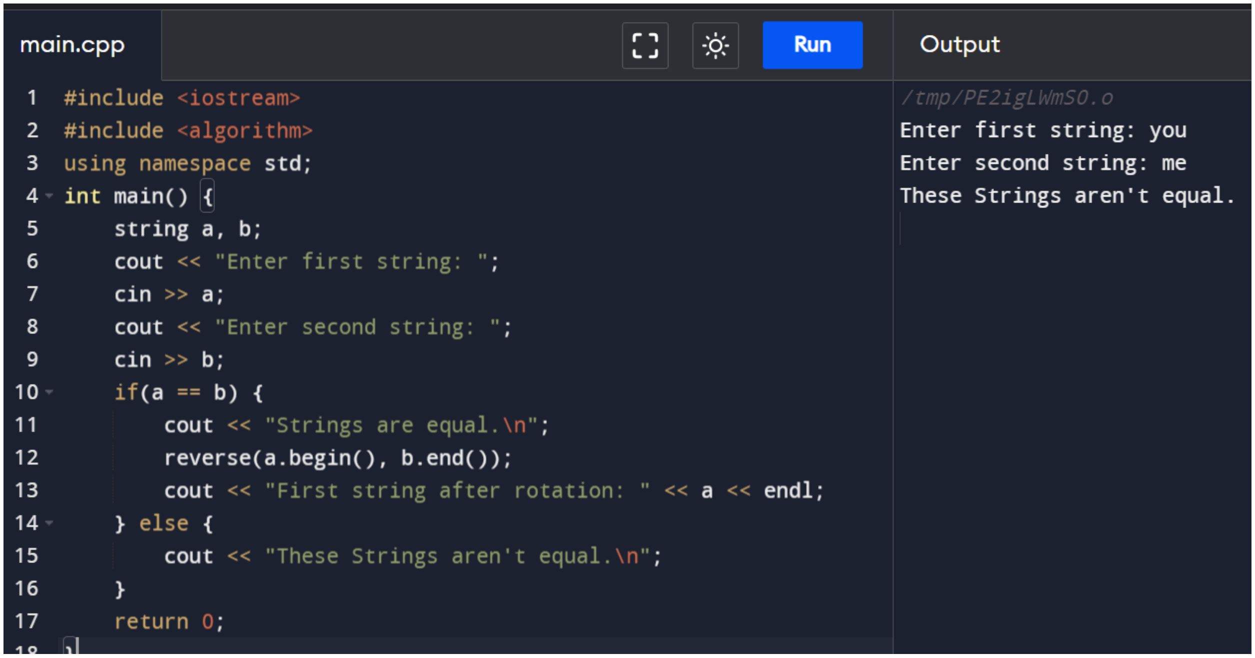


```
#include <iostream>
using namespace std;
bool Prime(int a) {
    if (a <= 1) {
        return false;
    }
    for (int b = 2; b * b <= a; b++) {
        if (a % b == 0) {
            return false;
        }
    }
    return true;
}
int main() {
    int N;
    cout << "Enter a number: ";
    cin >> N;
    int largest_Prime = N - 1;
```

```
/tmp/PE2igLWmS0.o
Enter a number: 78
The largest prime number less than or equal to 78 is 73
```

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 a, b;
   cout << "Enter first string: ";
   cin >> a;
   cout << "Enter second string: ";
   cin >> b;
   if(a == b) {
      cout << "Strings are equal.\n";
      reverse(a.begin(), b.end());
      cout << "First string after rotation: " << a << endl;
   } else {
      cout << "These Strings aren't equal.\n";
   }
   return 0;
}
```

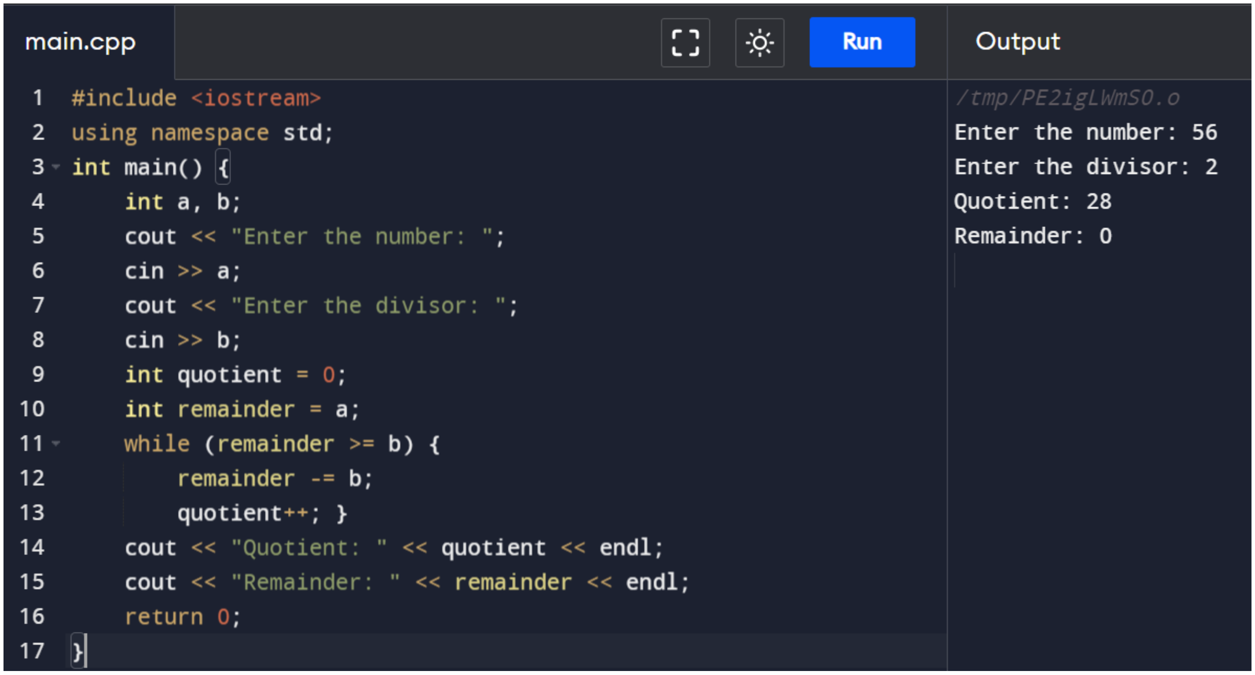


```
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    string a, b;
    cout << "Enter first string: ";
    cin >> a;
    cout << "Enter second string: ";
    cin >> b;
    if(a == b) {
        cout << "Strings are equal.\n";
        reverse(a.begin(), b.end());
        cout << "First string after rotation: " << a << endl;
    } else {
        cout << "These Strings aren't equal.\n";
    }
    return 0;
}
```

```
/tmp/PE2igLWmS0.o
Enter first string: you
Enter second string: me
These Strings aren't equal.
```

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 a, b;
    cout << "Enter the number: ";
    cin >> a;
    cout << "Enter the divisor: ";
    cin >> b;
    int quotient = 0;
    int remainder = a;
    while (remainder >= b) {
        remainder -= b;
        quotient++; }
    cout << "Quotient: " << quotient << endl;
    cout << "Remainder: " << remainder << endl;
    return 0;
}
```

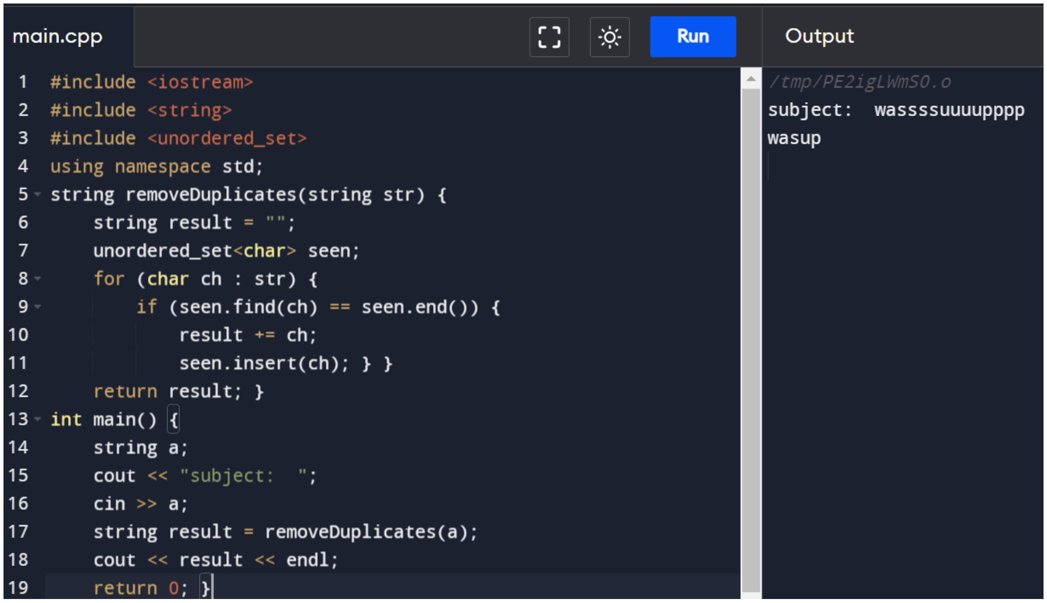


```
#include <iostream>
using namespace std;
int main() {
    int a, b;
    cout << "Enter the number: ";
    cin >> a;
    cout << "Enter the divisor: ";
    cin >> b;
    int quotient = 0;
    int remainder = a;
    while (remainder >= b) {
        remainder -= b;
        quotient++; }
    cout <<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

```
/tmp/PE2igLWmS0.o
Enter the number: 56
Enter the divisor: 2
Quotient: 28
Remainder: 0
```

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 a;
    cout << "subject:  ";
    cin >> a;
    string result = removeDuplicates(a);
    cout << result << endl;
    return 0; }
```

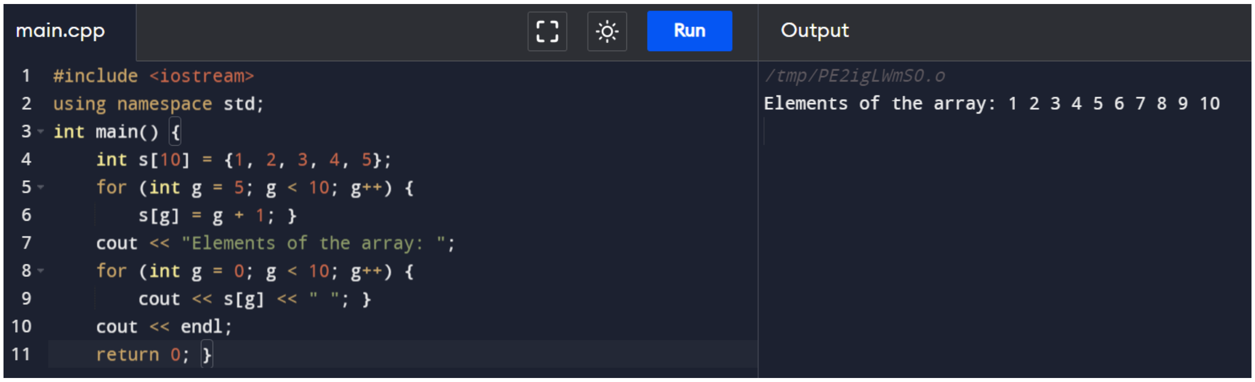


```
#include <iostream>
#include <string>
#include <unordered_set>
using namespace std;
string removeDuplicates(string str) {
    string result = "";
    unordered_set<char> seen;
    for (char ch : str) {
        if (seen.find(ch) == seen.end()) {
            result += ch;
            seen.insert(ch);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

```
/tmp/PE2igLWmS0.o
subject:  wassssuuuupppp
wasup
```

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

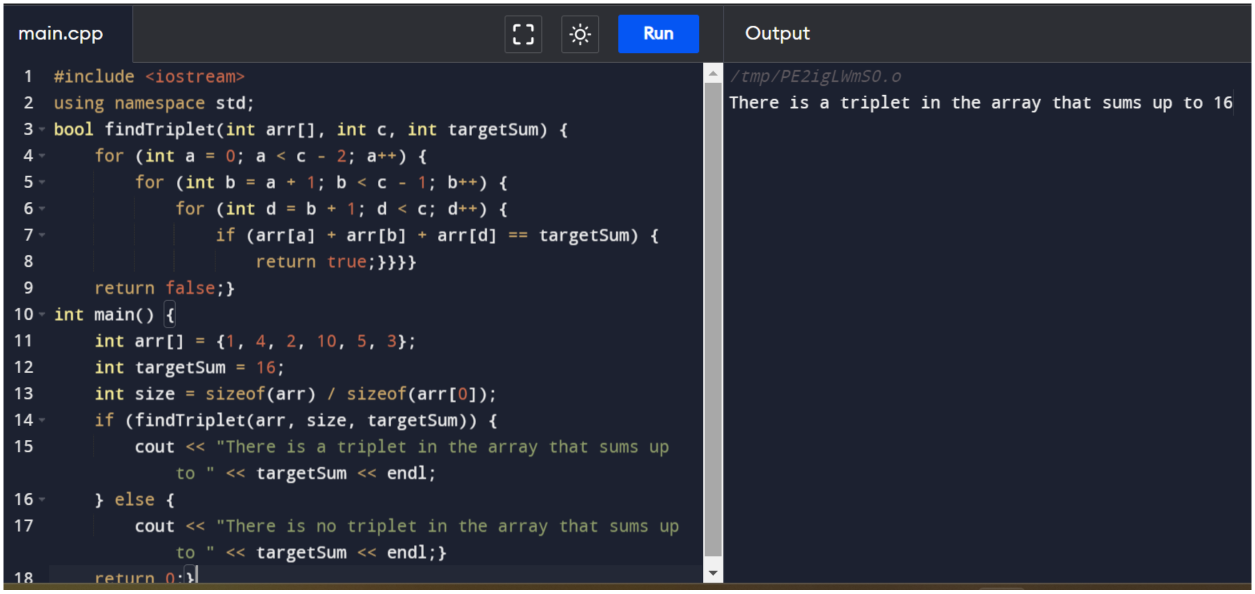


```
#include <iostream>
using namespace std;
int main() {
    int s[10] = {1, 2, 3, 4, 5};
    for (int g = 5; g < 10; g++) {
        s[g] = g + 1; }
    cout << "Elements of the array: ";
    for (int g = 0; g < 10; g++) {
        cout << s[g] << " "; }
    cout << endl;
    return 0; }
```

```
/tmp/PE2igLWmS0.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 c, int targetSum) {
   for (int a = 0; a < c - 2; a++) {
      for (int b = a + 1; b < c - 1; b++) {
         for (int d = b + 1; d < c; d++) {
            if (arr[a] + arr[b] + arr[d] == 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;}
```

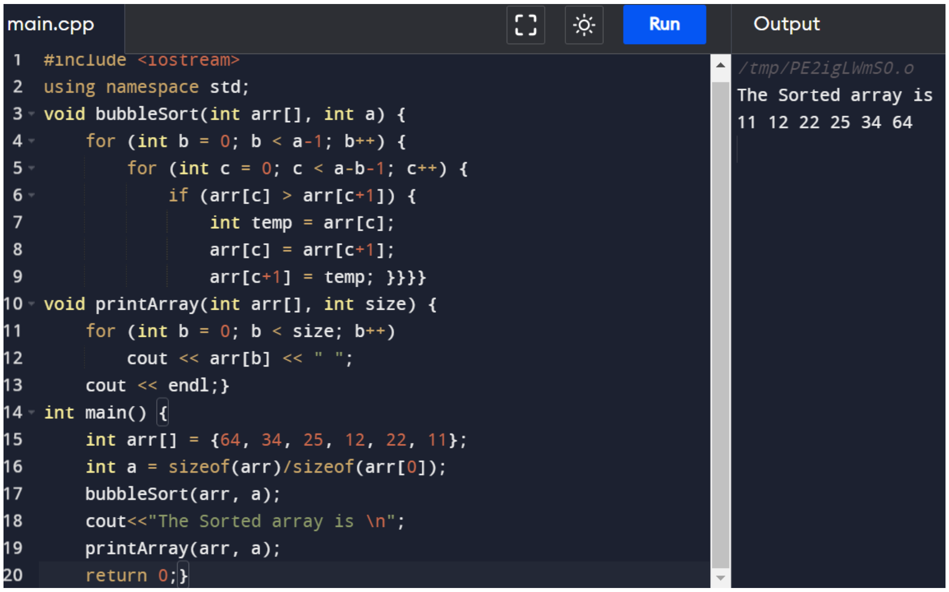


```
#include <iostream>
using namespace std;
bool findTriplet(int arr[], int c, int targetSum) {
    for (int a = 0; a < c - 2; a++) {
        for (int b = a + 1; b < c - 1; b++) {
            for (int d = b + 1; d < c; d++) {
                if (arr[a] + arr[b] + arr[d] == targetSum) {
                    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/PE2igLWmS0.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 a) {
    for (int b = 0; b < a-1; b++) {
        for (int c = 0; c < a-b-1; c++) {
            if (arr[c] > arr[c+1]) {
                int temp = arr[c];
                arr[c] = arr[c+1];
                arr[c+1] = temp; }}}}
void printArray(int arr[], int size) {
    for (int b = 0; b < size; b++)
        cout << arr[b] << " ";
    cout << endl;}
int main() {
    int arr[] = {64, 34, 25, 12, 22, 11};
    int a = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, a);
    cout<<"The Sorted array is \n";
    printArray(arr, a);
    return 0;}
```



```
#include <iostream>
using namespace std;
void bubbleSort(int arr[], int a) {
    for (int b = 0; b < a-1; b++) {
        for (int c = 0; c < a-b-1; c++) {
            if (arr[c] > arr[c+1]) {
                int temp = arr[c];
                arr[c] = arr[c+1];
                arr[c+1] = temp; }}}}
void printArray(int arr[], int size) {
    for (int b = 0; b < size; b++)
        cout << arr[b] << " ";
    cout << endl;}
int main() {
    int arr[] = {64, 34, 25, 12, 22, 11};
    int a = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, a);
    cout<<"The Sorted array is \n";
    printArray(arr, a);
    return 0;}
```

```
/tmp/PE2igLWmS0.o
The Sorted array is
11 12 22 25 34 64
```