

COMPUTER SCIENCE : C++

C++ Program to Check Whether a Number is Prime or Not

Example to check whether an integer (entered by the user) is a prime number or not using for loop and if...else statement.

To understand this example, you should have the knowledge of the following [C++ programming](#) topics:

- [C++ if, if...else and Nested if...else](#)
- [C++ for Loop](#)
- [C++ break and continue Statement](#)

A positive integer which is only divisible by 1 and itself is known as prime number.

For example: 13 is a prime number because it is only divisible by 1 and 13 but, 15 is not prime number because it is divisible by 1, 3, 5 and 15.

Check Prime Number

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

void main() {
    int n, i;
    bool isPrime = true;

    cout << "Enter a positive integer: ";
    cin >> n;

    for (i = 2; i <= n / 2; ++i) {
        if (n % i == 0) {
            isPrime = false;
            break;
        }
    }
    if (isPrime)
        cout << n << " is a prime number";
    else
        cout << n << " is not a prime number";

}
```

Output

```
Enter a positive integer: 29
29 is a prime number.
```

This program takes a positive integer from user and stores it in variable n .

Then, `for` loop is executed which checks whether the number entered by user is perfectly divisible by i or not.

The `for` loop initiates with an initial value of i equals to 2 and increasing the value of i in each iteration.

If the number entered by user is perfectly divisible

by i then, *isPrime* is set to `false` and the number will not be a prime number.

But, if the number is not perfectly divisible by i until test

condition $i \leq n/2$ is true means, it is only divisible by 1 and that number itself.

So, the given number is a prime number.

C++ Program to Check Whether a Number can be Express as Sum of Two Prime Numbers

Example to check if an integer (entered by the user) can be expressed as the sum of two prime numbers of all possible combinations with the use of functions.

To understand this example, you should have the knowledge of the following [C++ programming](#) topics:

- [C++ for Loop](#)
- [C++ if, if...else and Nested if...else](#)
- [C++ Functions](#)
- [Types of User-defined Functions in C++](#)

This program takes a positive integer from user and checks whether that number can be expressed as the sum of two prime numbers.

If the number can be expressed as sum of two prime numbers, the output shows the combination of the prime numbers.

To perform this task, a user-defined function is created to [check prime number](#).

Check Whether a Number can be Expressed as a Sum of Two Prime Numbers

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

bool checkPrime(int n);

void main()
{
    int n, i;
    bool flag = false;

    cout << "Enter a positive integer: ";
    cin >> n;

    for(i = 2; i <= n/2; ++i)
    {
        if (checkPrime(i))
        {
            if (checkPrime(n - i))
            {
                cout << n << " = " << i << " + " << n-i << endl;
                flag = true;
            }
        }
    }

    if (!flag)
        cout << n << " can't be expressed as sum of two prime numbers.";
}

// Check prime number
bool checkPrime(int n)
{
    int i;
    bool isPrime = true;

    for(i = 2; i <= n/2; ++i)
    {
        if(n % i == 0)
```

```
        {
            isPrime = false;
            break;
        }
    }

    return isPrime;
}
```

Output

```
Enter a positive integer: 34
34 = 3 + 31
34 = 5 + 29
34 = 11 + 23
34 = 17 + 17
```

C++ Program to Display Prime Numbers Between Two Intervals

Example to print all prime numbers between two numbers (entered by the user) in C++ Programming. This problem is solved using nested for loop and if...else statement.

To understand this example, you should have the knowledge of the following [C++ programming](#) topics:

- [C++ if, if...else and Nested if...else](#)
- [C++ for Loop](#)
- [C++ break and continue Statement](#)

Display Prime Numbers Between two Intervals

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

void main()
{
    int low, high, i, flag;

    cout << "Enter two numbers(intervals): ";
    cin >> low >> high;

    cout << "Prime numbers between " << low << " and " << high << " are: ";

    while (low < high)
    {
        flag = 0;

        for(i = 2; i <= low/2; ++i)
        {
            if(low % i == 0)
            {
                flag = 1;
                break;
            }
        }

        if (flag == 0)
            cout << low << " ";

        ++low;
    }
}
```

Output

```
Enter two numbers(intervals): 20 50 Prime numbers between 20 and 50 are: 23 29 31
37 41 43 47
```

In this program, the while loop is iterated $(high - low - 1)$ times.

In each iteration, whether `low` is a prime number or not is checked and the value of `low` is incremented by 1 until `low` is equal to `high`.

Visit this page to learn more on how to [check whether a number is prime or not](#).

If the user enters larger number first, this program doesn't work as intended. You can solve this issue by [swapping the numbers](#) if the user enters larger number first.

Display Prime Numbers When Larger Number is Entered first

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

void main()
{
    int low, high, flag, temp;

    cout << "Enter two numbers(intervals): ";
    cin >> low >> high;

    //swapping numbers if low is greater than high
    if (low > high) {
        temp = low;
        low = high;
        high = temp;
    }
}
```



```
cout << "Prime numbers between " << low << " and " << high << " are: ";

while (low < high)
{
    flag = 0;

    for(int i = 2; i <= low/2; ++i)
    {
        if(low % i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0)
        cout << low << " ";

    ++low;
}
}
```