```c
#include <stdio.h>

int main() {

  int n, i, flag = 0;
  printf("Enter a positive integer: ");
  scanf("%d", &n);

  // 0 and 1 are not prime numbers
  // change flag to 1 for non-prime number
  if (n == 0 || n == 1)
    flag = 1;

  for (i = 2; i <= n / 2; ++i) {

    // if n is divisible by i, then n is not prime
    // change flag to 1 for non-prime number
    if (n % i == 0) {
      flag = 1;
      break;
    }
  }

  // flag is 0 for prime numbers
  if (flag == 0)
    printf("%d is a prime number.", n);
  else
    printf("%d is not a prime number.", n);
```

Run Code

Output

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

In the program, a `for` loop is iterated from `i = 2` to `i < n/2`.

In each iteration, whether `n` is perfectly divisible by `i` is checked using:

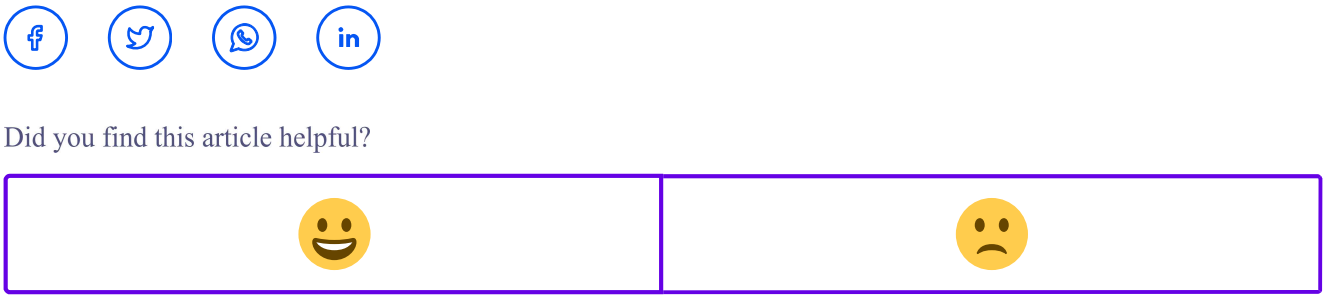
```
if (n % i == 0) {
  flag = 1;
  break;
}
```

If `n` is perfectly divisible by `i`, `n` is not a prime number. In this case, `flag` is set to **1**, and the loop is terminated using the `break` statement.

Notice that we have initialized `flag` as **0** during the start of our program.

So, if `n` is a prime number after the loop, `flag` will still be **0**. However, if `n` is a non-prime number, `flag` will be **1**.

Visit this page to learn how you can print all the prime numbers between two intervals.



Related Examples

C Example

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

C Example

Display Prime Numbers Between Intervals Using Function

C Example

Display Prime Numbers Between Two Intervals

C Example

Check Prime or Armstrong Number Using User-defined Function