C Program To Print Prime Numbers In A Given Range

In this tutorial, we will write a program to print prime numbers in a given range in C language. If you have to find all the prime numbers between some given ranges for example (39 to 48); to perform this task you will need a loop which will start from the starting range and ends at an ending range, and you will need a function which will check that the given number is prime or not as shown in the figure below.

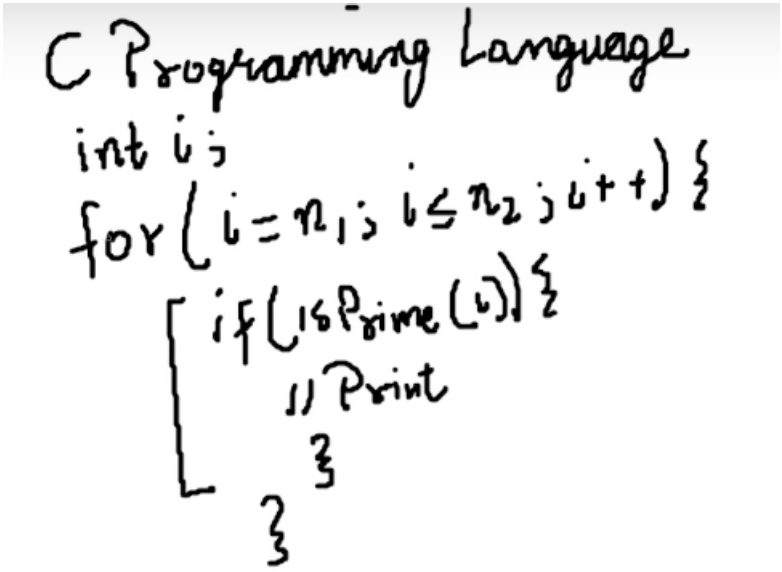

Figure 1: Find Prime Number

As shown in figure 1, we have a “for” loop which will start from the “n1” and ends at “n2”; “n1” is our starting range and “n2” is our ending range. The “if” condition inside the loop will call the function “isPrime”; the function “isPrime” will check if the given number is prime or not.

To determine whether a number is a prime number or not we have to check from (2 to n-1) that if that number is divisible with some number. If the number is divisible then it will not be a prime number but if the number is not divisible by any number from (2 to n-1) then the number is a prime number.

Note: But the main thing to note here is that we can also check whether the number is prime or not by checking the number form (2 to $\sqrt{n}$) rather than checking from (2 to n-1) to minimize the execution time.

An example program to print prime number in a given range in C language is shown below

```
int isPrime(int n){
    for (int i = 2; i*i <= n; i++)
    {
        if (n%i == 0){
            return 0;
        }
    }
    return 1;
}
```

Code Snippet 1: isPrime Function

As shown in code snippet 1, we have written a function that will check whether a number is a prime number or not.

1. The function takes 1 integer parameter “n”; the parameter “n” will be the number which will be checked whether it is a prime number or not.
2. We have used the “for” loop which will traverse from (2 to $\sqrt{n}$); at each iteration, the value of the variable “n” will be checked if it is divisible by some number.

3. If the value of the variable “n” is not divisible by any number from (2 to √n) then the function will return “1”; otherwise the function will return “0”

```
int main(){

    int n1, n2;

    printf("Enter first number\n");

    scanf("%d", &n1);

    printf("Enter second number\n");

    scanf("%d", &n2);
```

Code Snippet 2: Main Program 1

As shown in Code snippet 2,

1. We have declared two integer variables “n1” and “n2”; these variables will be used to store the input from the user.
2. The function “printf” is used to print “Enter first number” and the function “scanf” is used to get input from the user which will be stored in the variable “n1”
3. The function “printf” is used to print “Enter second number” and the function “scanf” is used to get input from the user which will be stored in the variable “n2”

```
printf("The prime numbers between %d and %d are: ",n1, n2);

    for (int i = n1; i <= n2; i++)

    {

        if(isPrime(i)){

            printf("%d ", i);

        }
```

```
    }

    return 0;

}
```

Code Snippet 3: Main Program 2

As shown in a code snippet 3,

1. The function “printf” is used to print “The prime numbers between %d and %d are:”; the main thing to note here is that at the place of “%d” the value of the variables “n1” and “n2” will be printed
2. The “for” loop is used to traverse from the starting range till the ending range; the variable “n1” is the starting number and the variable “n2” is the ending number.
3. In the “for” loop body the function “isprime” is called inside the “if” condition. The number will be printed when the “if” conditions get true

The output of the following program is shown in the figure below

Figure 2: Program Output

As shown in figure 2, we have given the number “3” as input for the first number and the number “29” as input for the second number. In the output, all the numbers between “3” and “29” which were prime numbers get printed.

Previous Next

CodeWithHarry