Java Concurrent Programming

CS331: Programming Languages Laboratory

Lab-1 Q6 Report

Task:

Finding sum of primes using multithreading

Author:

Gautam Sharma, 210101042

Instructor:

Prof. Sukumar Nandi, Dept. of CSE, IITG

Contents

1 Introduction

This lab was the first of the **11 labs** in the course **CS331: Programming Languages Laboratory**. This was an introductory lab, with **Multithreading**, and **Synchronization** in Java as the main topics being touched upon.

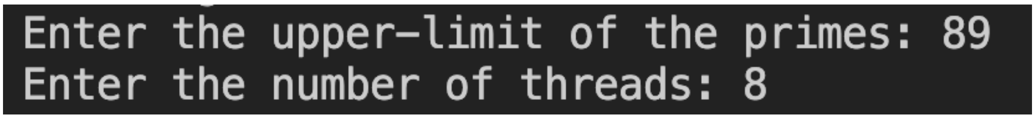
2 Task

To find the sum of primes less than the upper limit specified by the user.

3 Input

The user has to enter **two things**:

- The `upper-limit` for the primes.
  The **`size`** should be less than x?
- The number of threads (`numThread`) for the task.
  The number of threads **must be less than or equal to** the `upper-limit` and also must be less than or equal to `500000000`.

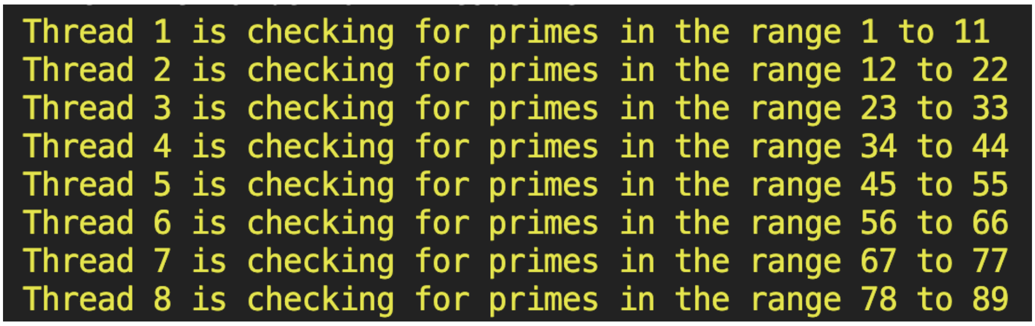


```
Enter the upper-limit of the primes: 89
Enter the number of threads: 8
```

Figure 1: Sample Input

4 Approach

The range [1, `upper-limit`] has been divided into `numThread` blocks of roughly equal size. Each thread looks for primes in one block.



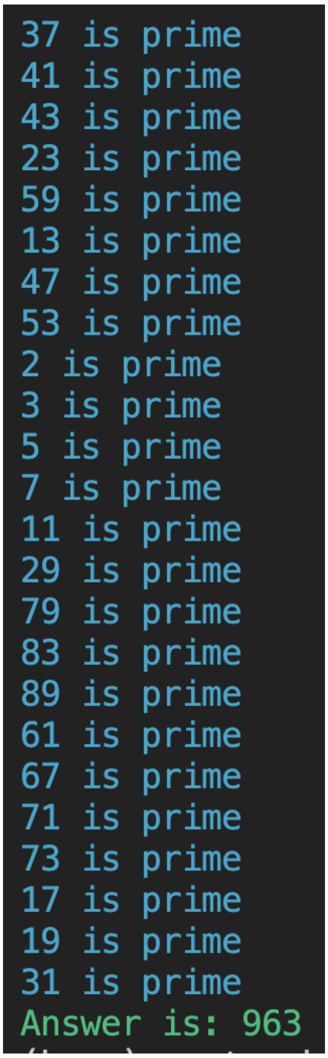
```
Thread 1 is checking for primes in the range 1 to 11
Thread 2 is checking for primes in the range 12 to 22
Thread 3 is checking for primes in the range 23 to 33
Thread 4 is checking for primes in the range 34 to 44
Thread 5 is checking for primes in the range 45 to 55
Thread 6 is checking for primes in the range 56 to 66
Thread 7 is checking for primes in the range 67 to 77
Thread 8 is checking for primes in the range 78 to 89
```

Figure 2: Sample processing

5 Output

There are **two outputs**:

1. List of primes less than or equal to `upper-limit`.
2. Sum of primes less than or equal to `upper-limit`.



```
37 is prime
41 is prime
43 is prime
23 is prime
59 is prime
13 is prime
47 is prime
53 is prime
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
29 is prime
79 is prime
83 is prime
89 is prime
61 is prime
67 is prime
71 is prime
73 is prime
17 is prime
19 is prime
31 is prime
Answer is: 963
```

Figure 3: Sample output

6 Technical Documentation

The task was implemented in the class `Q6` of the file `Q6.java`.

6.1 Public Variables

There are no public variables!

6.2 Private Variables

6.2.1 SUM

Type: `static`, `int`
Usage: Stores the sum of all primes less than `upper-limit`.

6.2.2 ANSI_RESET

Type: `constant`, `static`, `String`
Usage: Resetting output color to white

6.2.3 ANSI_RED

Type: `constant`, `static`, `String`
Usage: Changing output color to red

6.2.4 ANSI_GREEN

Type: `constant`, `static`, `String`
Usage: Changing output color to green

6.2.5 ANSI_YELLOW

Type: `constant`, `static`, `String`
Usage: Changing output color to yellow

6.2.6 ANSI_CYAN

Type: `constant`, `static`, `String`
Usage: Changing output color to cyan

6.3 Public Classes

There are no public classes!

6.4 Private Classes

6.4.1 prime

Type: Implements `Runnable`
Attributes: `l` - left-hand side of the range on which the thread operates, `r` - right-hand side of the range on which the thread operates, `sum` - stores the sum of primes in the range [l, r]
Usage: Spawns threads to find primes in a range

6.5 Public Methods

6.5.1 main

Type: `static`, `void`
Arguments: None
Usage: Taking the input and solving the task by calling the private methods.

6.6 Private Methods

6.6.1 add

Type: `void`, `static`, `synchronized`
Arguments: `int n`
Usage: Adds the answer calculated by each thread to the global variable SUM. The method is **synchronized**, that is it can be accessed by at most one thread at a time.

6.6.2 show_ans

Type: `void`
Arguments: None
Usage: Prints the global variable SUM to the user.

7 Program flow

The program undergoes the following steps:

1. Enters the `main` method, takes input and makes `numThread` number of threads.

2. Enters the `run` method of the `prime` class, which spawns a thread to find the number of primes in the range of the thread and finally calls the `add` method to add the sum of primes to SUM.

3. Final output is printed.

8 Usage

Refer to the README.