

Sieve of Eratosthenes

Introduction

A *prime number* is any integer that can be divided evenly only by itself and 1. The Sieve of Eratosthenes is a method of generating consecutive primes not exceeding any given integer $n > 1$. It was probably invented in ancient Greece and is known as the sieve of Eratosthenes (ca. 200 b.c.). The initial algorithm of Eratosthenes starts by initializing a list of prime candidates with consecutive integers from 2 to n . Then, on its first iteration, the algorithm eliminates from the list all multiples of 2, i.e., 4, 6, and so on. Then it moves to the next item on the list, which is 3, and eliminates its multiples. No pass for number 4 is needed: since 4 itself and all its multiples are also multiples of 2, they were already eliminated on a previous pass. The next remaining number on the list, which is used on the third pass, is 5. The algorithm continues in this fashion until no more numbers can be eliminated from the list.

Problem Statement

Create a Java classes (**Sieve.java**) that uses the Sieve of Eratosthenes method to generate and print the prime numbers between 2 and n . Your program must use the **ABQueue.java** and the **CirQueue.java** classes that we have developed in class.

Problem Analysis

In this program, we will use a variation of the algorithm that utilizes queues instead of arrays. Create a circular queue of integers and fill it with the consecutive integers 2 through n , inclusive. You also need to create an empty simple queue to store the prime numbers. Then, you obtain the next prime number p by removing it from the circular and placing it in the primes queue. You then need to go through the circular queue and eliminate all numbers that are divisible by the prime number p . Repeat this process until p reaches \sqrt{n} . All remaining numbers in the circular queue must be prime, by now. So, all you have to do is to transfer them to the queue of primes.

Program Input

A positive integer $n \geq 2$. Your program should reject number less than 2.

Program Output

The program must display all prime numbers between 2 and n . Display only 10 prime numbers on each line of output, except for the last line of output which could have less than 10 numbers. Your program must then inquire whether the user wants to run it again and act accordingly. Be specific with your prompts.

Submission Instructions

When you are satisfied with your implementation, and after testing it thoroughly, submit the **Sieve.java** file via Blackboard Learn.