|  |  |  |
| --- | --- | --- |
| **AP Computer Science** | **TextLab06 Java Assignment** | |
| **The "Sieve of Eratosthenes" Program** | | **80 & 100 Point Versions** |
| **Assignment Purpose:**  The purpose of this assignment is to practice declaring one-dimensional array objects and manipulating the elements of the array. | | |

Write a program that computes prime numbers using the “Sieve of Eratosthenes” method.

The **Sieve** prime number generator uses an ingenious method, which does not involve any type of division, by using the following steps:

[1] Initialize all numbers in the array, starting with 2, as primenumbers. Ignore number 1.

[2] Check the first number, **2**, to see if it is prime.

Since it is designated prime, change all the multiples of 2 to **Not Prime**.

[3] Check the next number, **3**, to see if it is prime.

Since it is designated prime, change all the multiple of 3 to **Not Prime**.

[4] Continue this process, until the upper limit is reached.

Imagine that a small upper limit of 21 is requested.

The “Sieve” will work with **Pr** (**Pr**ime) and **NP** (**N**ot **P**rime) as follows:

**STEP 01** Initialize all elements to **Prime**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xx | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr | Pr |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

**STEP 02** Change all multiples of **2** to **Not Prime**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xx | Pr | Pr | NP | Pr | NP | Pr | NP | Pr | NP | Pr | NP | Pr | NP | Pr | NP | Pr | NP | Pr | NP | Pr |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

**STEP 03** Change all multiples of **3** to **Not Prime**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xx | Pr | Pr | NP | Pr | NP | Pr | NP | NP | NP | Pr | NP | Pr | NP | NP | NP | Pr | NP | Pr | NP | NP |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

**STEP 04** Repeat this process until the upper limit is reached

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xx | **Pr** | **Pr** | NP | **Pr** | NP | **Pr** | NP | NP | NP | **Pr** | NP | **Pr** | NP | NP | NP | **Pr** | NP | **Pr** | NP | NP |
| 1 | **2** | **3** | 4 | **5** | 6 | **7** | 8 | 9 | 10 | **11** | 12 | **13** | 14 | 15 | 16 | **17** | 18 | **19** | 20 | 21 |

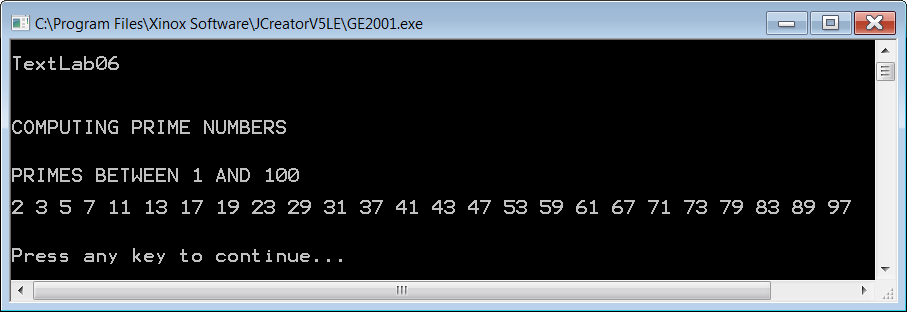
**Prime Numbers left are: 2, 3, 5 ,7 , 11, 13, 17, 19**

**80 Point Version Specifics**

The 80-point version displays all the prime numbers between **1** and **100**. Complete methods **ComputePrimes** and **DisplayPrimes** inside the **Lab12st** class. There is only a single execution and there is no program user input at all.

|  |  |
| --- | --- |
| **TextLab06 Student Version** | **Do not copy this file, which is provided.** |
| public static void main(String args[])  {  // This main method needs additions for the 100 point version.  Scanner input = new Scanner(System.in);  System.out.println("\nTextLab06\n");  final int MAX = 100;  boolean primes[];  primes = new boolean[MAX];  computePrimes(primes);  displayPrimes(primes);  } | |

**80 Point Version Output (Only 1 required)**



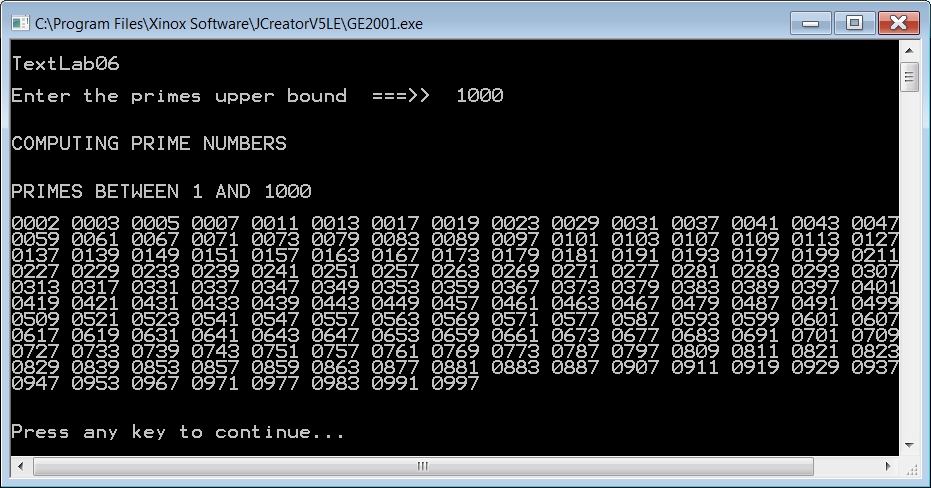
**100 Point Version Specifics**

The 100-point version requires interactive input in a text window. Additionally, the 100-point version needs to format program output so that all prime numbers display four digit numbers with leading zeroes where necessary using a **DecimalFormat** of **"0000"**. To make the output look proper 1 blank space needs to be printed after each number. Execute the program twice.

|  |  |
| --- | --- |
| **TextLab06 100 Point Version** | **Required main Method** |
| public static void main(String args[])  {  System.out.println("\nTextLab06\n");  Scanner input = new Scanner(System.in);  System.out.print("Enter the primes upper bound ===>> ");  final int MAX = input.nextInt();  boolean primes[] = new boolean[MAX];  computePrimes(primes);  displayPrimes(primes);  } | |

**100 Point Version Outputs (2 required)**

***First Output***



***Second Output***

