Parallel Programming

1) write a program that uses multiple threads to solve the problem of finding the integer in the range 1 to 10000 that has the largest number of divisors, but for the range 1 to 100000. At the end of the program, output the elapsed time, the integer that has the largest number of divisors, and the number of divisors that it has.

2) We need to write a function called frequency. It takes a slice of strings and a worker count as parameters. The return value is a hashmap. The keys are all the letters those strings contain, their value the amount of times that letter appears. This needs to be done in worker_count number of process.

3) Implement three classes: Storage, Counter, and Printer. The Storage class should store an integer. The Counter class should create a thread that starts counting from 0 (0, 1, 2, 3 ...) and stores each value in the Storage class. The Printer class should create a thread that keeps reading the value in the Storage class and printing it. Write a program that creates an instance of the Storage class and sets up a Counter and a Printer object to operate on it.

4) Write a program using multi-threads in which create a global integer vector. function FibonacciNumber(int, int) that takes 02 integers to calculate the next number in Fibonacci series and store it in the vector. In main create 02 threads and call assign the FinbonacciNumber function to each of the thread to calculate the next number and store the result in the integer vector. another thread that prints the Fibonacci series in parallel. FibonacciNumber(int n1, int n2); //function to calculate the next number in Fibonacci series. printFibSeries() const; //function to print the current status of series.

5) Write a multi processing program in which the first process prints odd numbers between 100 to 200. Second Process print the prime number. Between 200 to 300. And third process generate Armstrong number between 100 to 300.

6) Write a Python program to define a subclass using threading and instantiate the subclass and trigger the thread which implement the task of building dictionary for each character from the the given input string.

7) Write a program Find that searches all files specified on the command line and prints out all lines containing a reserved word. Start a new thread for each file.

8) Implement the merge sort algorithm by spawning a new thread for each smaller MergeSorter. Hint: Use the join method of the Thread class to wait for the spawned threads to finish.