## Technical Assignment

Please submit the solution for this assignment in a (.zip) file with only the source code files, no executables (.exe/.dll/...) otherwise, your email will be blocked automatically by the anti-virus scanner.

This assignment should be delivered by the time indicated in the email sent to you. The programming language is C# **(We understand that C# is not your preferred language, we want you to learn as you go, because we learn as we go in our work)**

**Getting Ready**

1. Download Visual Studio (https://visualstudio.microsoft.com/downloads/) or Visual Studio Code (https://code.visualstudio.com/#alt-downloads).
2. The C# language reference is here: https://docs.microsoft.com/en-us/dotnet/csharp/

**What you need to know/learn before you start**

1. The C# programming language.
2. Writing console application.
3. The `System.Threading` namespace.
4. The `System.Concurrent` namespace.

**The assignment**

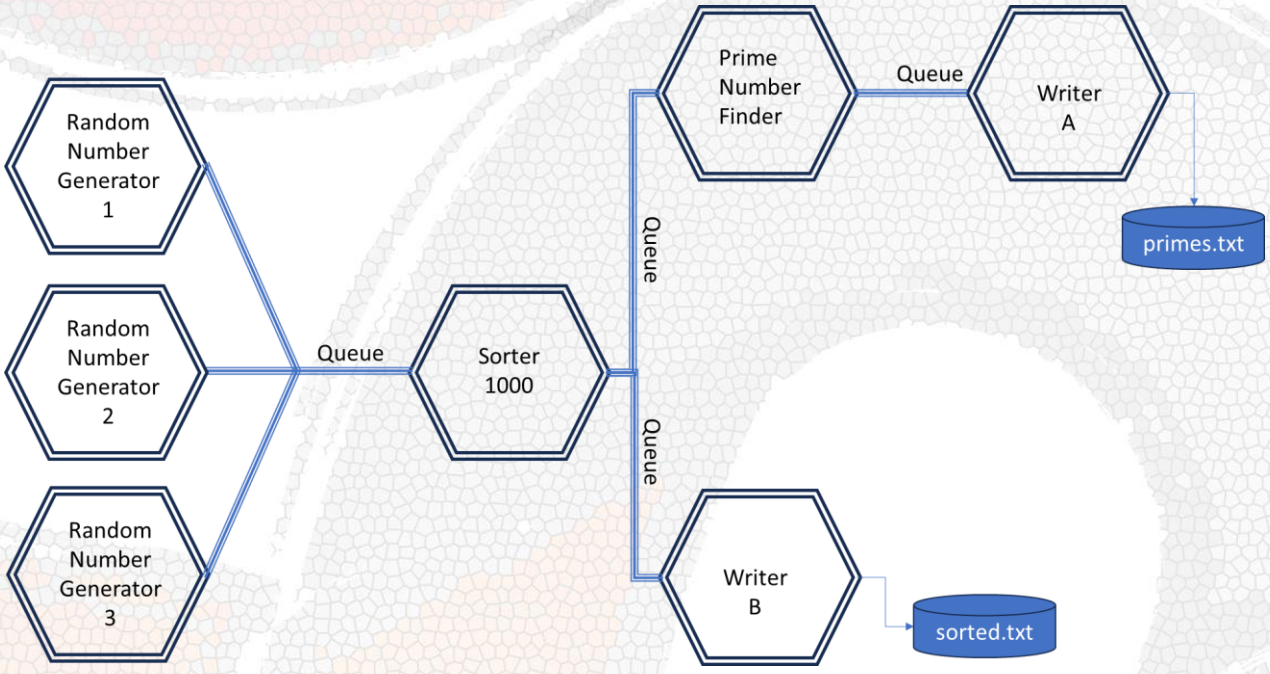


*Figure 1 System diagram*

Implement the system above which consists of the 7 parallel tasks running at the same time, each hexagon represents a self-contained running process/thread or task.

You should use the .net Task class for your implementation, you must also ensure data integrity and safety.

The table below describes each task and queue in the system.

| Element | Primary Function | Output |
|---|---|---|
| **Random Number Generator 1**<br>**Random Number Generator 2**<br>**Random Number Generator 3** | Generates random numbers, seeded from the system time. The random number pushes the numbers to the queue of the sorter. (We don't care if you use integers, floats or decimals). Make sure that each random number generator is seeded differently than the others, otherwise the same numbers will repeat.<br>You should push numbers as fast as possible with no delays. | Random numbers pushed to the sorter queue. |
| **Sorter 1000** | This sorter takes the data from the queue and when the number of elements taken reaches 1000 elements, it sorts them and send them to two queues (a copy for each queue).<br>One queue goes to the Prime Number Finder. The other queue goes to the Writer.<br>You should decide if:<br>1. How to sort the numbers as fast as possible.<br>2. Decide if you want to send them one number at a time, or a group of 1000 numbers. Find the fastest way. | 1000 sorted numbers as individual numbers or as a single group. One copy for each queue. |
| **Prime Number Finder** | This task will find prime numbers in the incoming numbers.<br>Any prime found, will be pushed to the output queue. | Produces a list of prime numbers. |
| **Writer A** | This task write the prime numbers (append) the file `primes.txt` on the disk.<br>You MUST show the source of the number (Random number generator 1,2 or 3)<br>Example:<br>131, RNG1<br>93, RNG2<br>.<br>. | Produces a file on disk `primes.txt`. |
| **Writer B** | This task writes the numbers (append) to the file `sorted.txt` on the disk.<br>You MUST show the source of the number (Random number generator 1,2 or 3)<br>Example:<br>1, RNG1<br>53, RNG2<br>56, RNG3<br>.<br>.<br>. | Produces a file on disk `sorted.txt` |

**Bonus requirements (5 stars)**

- Show the number of elements in the queues on the console.
- Show the memory usage of your system.
- Show the speed of writing to sorted.txt and primes.txt (example 100 numbers/second).

## How to pass the assignment?

1. Write fast code, we will test the speed of your code, we don't care if you finish early.
2. Write clear and simple code.
3. Submit on time, just the source code in a .zip file, don't include any .exe or .dll files, your email could be rejected by the automatic anti-virus.
4. Try to answer the optional bonus question.

## Questions?

If you have questions, send an email to hello@alephon.com. We can clarify the assignment by email, but we don't offer help solving it.