

To handle a large amount of data it has been decided to filter the data into separate files.

The data is larger than the local storage capacity so all the data cannot be handled at once.

So a number of simplifications have been made to the above scenario.

- \_ The data is assumed to be integers.
- \_ The size of the local storage capacity is  $m$  (below  $m$  is chosen to be 4).
- \_ The size of the data is  $n$  (below  $n$  is chosen to be  $100m = 400$ ).
- \_ Any storage needed to handle concurrency correctly is assumed to be negligible
- compared to  $n$  and  $m$ , so this is disregarded.
- \_ The filtering is into  $k$  files (below  $k$  is chosen to be 2).
- [Clearly, in a real application  $m$  and  $n$  would be several magnitudes larger.]

## Tips\* **Assignment Project Exam Help**

This program is a multithreaded program with three concurrent threads, A, B and C.

<https://powcoder.com>

Thread A generates  $n$  random integer numbers and writes them into a shared memory with a capacity of  $m$  integers. Due to the limited capacity of the shared memory, thread A cannot write

all numbers at once. Threads B and C read the integer numbers from the shared memory.

Thread B writes the even numbers to the file even-numbers and thread C writes the odd

numbers to the file odd-numbers. Every number generated by thread A should be written

in one of the two files. No number should be written twice.

\*Different values of  $m$ ,  $n$ , and  $k$

For  $k = 3$  you may consider the three classes of odd numbers, even numbers, and multiples

of 3. Note that these are non-disjoint: numbers belonging to more than one class can

be filtered into any of these but should still only appear in one of the output files.

**Examples of things you should explain:**

**Where in the code to define X?**

**How can you be sure that Y happens/does not happen?**

**What is the purpose of some identified piece of code?**

**How to solve the problem?**

**Assignment Project Exam Help**

**<https://powcoder.com>**

**Add WeChat powcoder**