CSE 314
Online (B2)
Time: 40 minutes

There is a buffer (i.e., an array) of integers named as *Storage* which is shared by 3 processes named as **Generator**, **OddFilter**, **EvenFilter**. The buffer has a fixed capacity of 5 and the initial buffer size is 0. There are also 2 expandable buffers of integers *OddNumber* and *EvenNumber*. *OddNumber* is accessed by **OddFilter** and *EvenNumber* is accessed by **EvenFilter**. You can consider each of these two buffers as a queue.

The processes work as follows:

1) If the buffer *Storage* is empty, then **Generator** generates 5 random numbers between 1 to 100, and inserts those numbers into the buffer *Storage*. When **Generator** accesses the buffer *Storage*, no other Processes can access the buffer. You have to print the generated numbers by **Generator**.
2) The Process **Generator** can generate total 50 numbers in 10 number of iterations. After 10 iterations, it sets a global variable *stop* = 1, and terminates. The Processes **OddFilter** and **EvenFilter** also terminate if the buffer *Storage* is empty and *stop* = 1. You have to print the buffers *OddNumber* and *EvenNumber* after all the Processes terminates.
3) **OddFilter** fetches the odd numbers from the buffer *Storage* and pushes those odd numbers to the buffer *OddNumber*.
4) **EvenFilter** fetches the even numbers from the buffer *Storage* and pushes those even numbers to the buffer *EvenNumber*.
5) Each time a number is fetched by either **OddFilter** or **EvenFilter** from *Storage*, the size of *Storage* is decreased by 1. If the buffer *Storage* is empty, both Processes **OddFilter** and **EvenFilter** wait.

*****You must implement **the three Processes as three (3) individual functions.** Write necessary program to implement the Processes with proper synchronization among them.

Sample Output: [1 iteration]

EvenFilter starts working…

Generator starts working…

Storage is empty……EvenFilter waits for numbers to be generated…

OddFilte starts working…

Generator generates: 10 25 3 42 79… and size of Storage = 5

OddFilter fetches 25… and size of Storage = 4

OddFilter fetches 3… and size of Storage = 3

Storage is not empty… Generator waits for Storage to be empty…

EvenFilter fetches 10… and size of Storage = 2

Generator terminates…

OddFilter fetches 79… and size of Storage = 1

OddFilter completes working…

EvenFilter processes… and size of Storage = 0

EvenFilter completes working…

EvenFilter terminates…

OddFilter terminates…

OddNumber contains: 25 3 79

EvenNumber contains: 10 42