Report of multi_pc.c (CS3500)

- **By Prakash A (CS20B061)**

The muli_pc.c contains the implementation of multiple producer and consumer problems , where each producer and consumer are implemented as thread . Use semaphores to synchronize between the producers and consumers. The items generated are using the rand()%500 function which gives a random value between 0 to 500.

All the producers and consumers will have a shared buffer which you can consider to be a stack like data structure. Whenever a producer produces an item (a number in our case), it is placed on top of the stack. And similarly, whenever a consumer wants to consume an item, it picks a number from the top of the stack. All the producers and consumers should be implemented as threads

The Question :
Suppose if you start the consumer threads, but put all producer threads to sleep for (say) 5 seconds, then will the output of your program change?

The Answer:
The output of the program won't change but the output gets executed in parts.
Because the sleep(5) just stops the thread from executing , but does not change the order of execution since it is present only in the producer thread.
The output may change depending on where you put the sleep(5) in the producer thread , if it is after the product is put in the buffer there is a chance it will change, otherwise it will not since the order of execution won't change.
And here “output change” is a vague term so cant be described correctly.