**Advanced Object Oriented Programming Laboratory**

**Offline Assignment**

1. **Divide and Conquer Sorting**

Suppose, you want to sort some numbers but you want to use multithreading for this. Any number of integers can be supplied to your program. Moreover, you can also provide X as input where X is the number of divisions of the array to sort. You will have to divide the array into X parts and sort them independently before receiving the entire output and then combine them into one sorted array.

Consider the array as a shared resource and the computation step as a shared method. So multiple threads shouldn’t be allowed to sort at the same time.

Model the division step as different threads and implement the scenario with proper **synchronization**.

Every thread must print a line in the console once it performs some activity. For example: “Thread t1 sorting array from index l to r”, where l and r would be the values of the left and right indices between which the thread is sorting the array.

1. **Producer and Consumer Problem**

Suppose you have an array of length 5. The producer produces a random positive number and places it at the (you might need to keep a counter to keep track of the front of the array). The producer sleeps for 2 seconds after it has placed something in the array. If the array is full (that is, all 5 positions of the array are occupied with a number), the producer will go into the waiting state. Every time the producer produces and keeps an element in the array, you need to print out a message.   
  
The consumer consumes elements from the front of the array. The consumer goes to sleep for 1 second after it has consumed something. If the array is empty, the consumer will go into the waiting state. Every time the consumer consumes an element from the array, you need to print out a message.   
  
You need to implement the above scenario with multithreading (the producer and consumer should run on two different threads), proper synchronization and calls to wait() and notify() where appropriate.