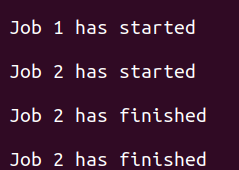
[21AIE202 Operating Systems (2022 Odd)](https://amritauniv.sharepoint.com/sites/21AIE202OperatingSystems2022Odd)

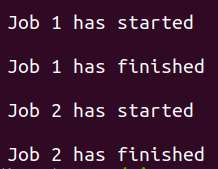
**Lab Exercise** 4

Thread Synchronization using Mutex  
Max marks: **6** / Time: **3 hours  
Submission Time: 11** PM, 11 Jan 2023

All the related files can be accessed from our Sharepoint course page.

Go through the thread\_sync\_1.c code and understand what the code does. Now, compile the code and run it. You should get an output as given below.

Here, you can see that the common variable counter is modified by the 2 threads (in any order), and hence can cause **race condition.**

Ideally, what we need is the first thread to execute 1st, complete the job, and then let the thread 2 start execution. In order to achieve this you have to modify the code given in thread\_sync\_2.c. Please read the comments given in the code and make changes to the code so as to ensure that the output looks like the one given below.

1. **Question [3 Marks]: Open and modify** thread\_sync\_3.c file, in which you will have to create two threads and modify and global variable counter (that is initialized to your last four digits of the roll number) using those two threads. The details of what needs to be coded is given as comments in the code. (Note that you are NOT using mutex in this code)  
     
   After modifying, compile and run the executable file to get the output.

<< COPY PASTE the formatted code below >>  
<< TAKE A SNIP of the output and paste it below >>

1. **Question [3 Marks]: Open and modify** thread\_sync\_3\_with\_mutex.c – In this file you are required to protect the critical sections using mutex. The details of what needs to be coded is given as comments in the code.   
     
   << COPY PASTE the formatted code below >>  
   << TAKE A SNIP of the output and paste it below >>