**Lab 10 Thread Synchronisation:**

**test and set;**

Q1 Run and Test the following two programs covered in lecture **: mutex2.c and mutex2\_lock.c Clearly explain** the output of each and how the locking preservers the integrity of the program: ensure it does not generate spurious outputs (Results)

**wait and signal**

Q 2 Run and test the wait\_signal.c program given in the class and take screen shots each test/run. Clearly explain the output of the program based on the waiting and signalling concepts covered in the class.

Q 3:

Modify Q2 the wait\_signal.c . This time create two more threads: thread 4 calls the Wait\_fnt and thread 5 calls the signal function

(do not forget to the declare two thread variables t4 =4 and t5 =5)

Run the program a number of times and insert screen shots into a word document. Clear explain the output.

Semaphores

Q4 Run the semaphore\_example.c covered in the class: insert screen shots of output into a word document. Explain the output.