CS 376	**Operating Systems**	Test 2
Name:		Maximum Points: 48

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages.

1. [10 pt] The following solution to the critical section problem was given by T. Dekker. The code given is for process p_1 and a similar code can be assumed to exist for process p_2.

```
#define TRUE 1
#define FALSE 0

int c1 = TRUE,               /* Boolean flag variable for process p_1     */
    c2 = TRUE,               /* Boolean flag variable for process p_2     */
    turn = 1;

while ( TRUE )
{
    c1 = FALSE;
    while ( ! c2 )
        if ( turn == 2 )
        {
            c1 = TRUE;
            while ( turn == 2 );
            c1 = FALSE;
        }

    critical_section_1;

    turn = 2;
    c1 = TRUE;
}
```

Show how this solution prevents mutual exclusion. Does it take care of the other two problems involved in the critical section problem?

2. [12 pt] Suppose the following jobs arrive for processing at the times indicated and run the specified amount of time. Calculate the average waiting time and average turnaround time for the scheduling disciplines listed.

Job	Arrival Time	Length
1	0.0	8
2	0.4	4
3	1.0	1
4	1.2	5
5	1.6	9

(a) FIFO

(b) SJN

(c) SJN; however, the CPU is left idle for the first time unit

(d) SRTF

(e) Round robin, with a quantum of 0.2

3. [15 pt] Consider the following snapshot of a system. There are no current outstanding queued unsatisfied requests.

	Current allocation				Maximum demand				Still needs			
Process	R_1	R_2	R_3	R_4	R_1	R_2	R_3	R_4	R_1	R_2	R_3	R_4
p_1	0	0	1	2	0	0	1	2				
p_2	2	0	0	0	2	7	5	0				
p_3	0	0	3	4	6	6	5	6				
p_4	2	3	5	4	4	3	5	6				
p_5	0	3	3	2	0	6	5	2				
Available	2	1	0	0								

(a) Compute what each process still might request and display in the columns labeled "still needs."

(b) Is this system currently in a safe or unsafe state? Why?

(c) Is this system currently deadlocked? Why or why not?

(d) Which processes, if any, are or may become deadlocked?

(e) If a request from p_3 arrives for $(0, 1, 0, 0)$, can that request be safely granted immediately? In what state (deadlocked, safe, unsafe) would immediately granting that whole request leave the system? Which processes, if any, are or may become deadlocked if this whole request is granted immediately?

4. [5 pt] Consider a system consisting of three processes and a SR resource class with 4 units. Each process needs at most two units. Is the system deadlock free (i.e., all states are safe)?

5. [6 pt] Consider a system with 4200 words of main memory using variable partitions. At some time, the memory will be occupied by three blocks of code/data as follows:

Starting Address	Length
1000	1000
2900	500
3400	800

When loading a new block into memory, the following strategy is used:

- Try the best-fit algorithm to locate a hole of appropriate size.
- If this fails, create a larger hole by shifting blocks in memory toward address zero; this always starts with the block currently at the lowest memory address and continues only until enough space is created to place the new block (partial compaction).

Assume that three new blocks with the respective sizes 500, 1200, and 200 are to be loaded (in the order listed). Show the memory contents after all three requests have been satisfied.