(a) Is this system deadlocked?

(b) Are there any blocked processes?

(c) What is the resulting graph after reduction by P_1?

(d) What is the resulting graph after reduction by P_2?

(e) Both P_1 and P_2 have requested R_2:

i. What is the status of the system if P_2's request is granted before P_1's?

ii. What is the status of the system if P_1's request is granted before P_2's?

4. [12pt] Consider a computing system with 13 tape drives. All jobs running on this system require a maximum of 5 tape drives to complete but they each run for long periods of time with just 4 drives and request the fifth one only at the very end of the run. The job stream is endless.

(a) If your operating system supports a very conservative device allocation policy, no job will be started unless all tapes requested have been allocated to it for the duration of its run:

i. What is the maximum number of jobs that can be active at once.

ii. What are the minimum and maximum number of tape drives that may be idle as a result of this policy?

iii. Explain your answer.

(b) If your operating system supports the Banker's algorithm:

i. What is the maximum number of jobs that can be active at once.

ii. What are the minimum and maximum number of tape drives that may be idle as a result of this policy?

iii. Explain your answer.

5. [6pt] Explain the following terms:

(a) Fragmentation

(b) Compaction

(c) Relocation

6. [6pt] Given the following information:

Job List		Memory list		
Job stream	Memory requested	Memory blocks	Size	Starting address
P_1	740K	Block 1	610K	0000K
P_2	500K	Block 2	850K	0610K
P_3	700K	Block 3	700K	1460K

Do the following:

(a) Use the best-fit algorithm to allocate the memory blocks to the three arriving jobs.

(b) Use the first-fit algorithm to allocate the memory blocks to the three arriving jobs.

CS 376 **Operating Systems** Test 2

Name: Maximum Points: 49

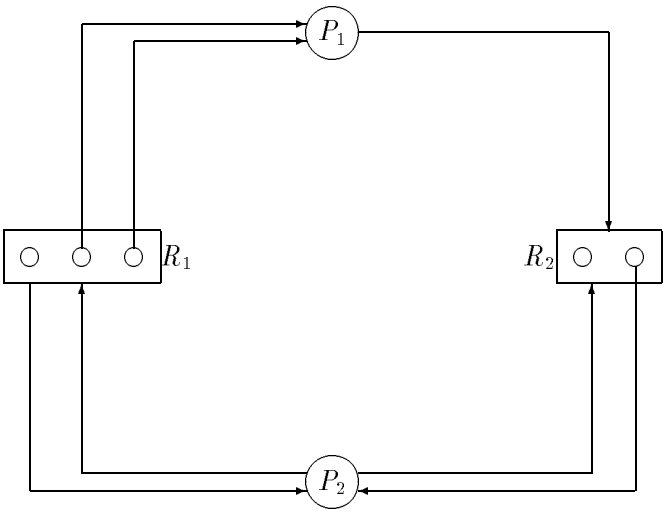
Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages.

1. [8pt] The ability of one process to spawn a new process is an important capability, but it is not without its dangers. Consider the consequences of allowing a user to run the following code (available in Unix):

```
function wildone ( void );
{
    while ( 1 )
        pid = fork();
}
```

   (a) What are the consequences of running such a program?

   (b) As per the basic theory of computation, it is impossible to predict a path of execution a program may take (halting problem). So, what controls can be built into the operating system to detect processes like this one at run time? Would the controls you propose hinder a process's ability to spawn new processes?

2. [5pt] Five jobs are in the ready queue waiting to be processed. Their estimated CPU time requirement is as follows: 10, 3, 5, 6, and 2. What order should they be processed using one of the non-preemptive scheduling algorithms to minimize average waiting time? What will be the average waiting time in that case? What will be the average turnaround time?

3. [12 pt] Consider the following directed resource allocation graph: