Name ______________________________ ID# ____________________

Sample Final

Name ______________________________________ ID# ____________________

PART A

Mark the following (T-F) questions using T for true, and F for false.
Write your answers in the table provided below (page 4).

1. There are solutions to synchronization problems that can be implemented using monitors, but cannot be implemented using semaphores.
2. Reentrant code is code that cannot be used by several users simultaneously, since it contains code that users can alter during execution.
3. Shared code in a segmentation scheme must have the same segment number for each user.
4. Throughput is the difference of time between submission of a job to its execution.
5. Global replacement states that the victim of replacement can be selected from any of the pages in the current process.
6. Thrashing is a phenomenon that occurs when the system spends excessive amount of time on DMA, compared to the execution of processes.
7. Privileged instructions cannot be executed in monitor mode..
8. To handle memory swaps while performing I/O, we can lock the pages waiting for I/O into memory using a lock bit.
9. Banker's Algorithm can be used for deadlock avoidance when there are multiple instances of each resource type.
10. When using a single directory, users have no privacy and they must be careful in choosing file names.
11. An interrupt vector contains the saved program counter values of interrupted user programs.
12. In a system with only one instance of each resource type, the presence of a cycle in the wait for graph is sufficient to detect a deadlock.

Name ______________________________________ ID# ____________________

13. The convoy effect occurs when we have a single CPU bound job and a number of I/O bound jobs.

14. In Solaris 2 the LWPs (Light Weight Processes) are the kernel threads.

15. There are solutions to synchronization problems that can be implemented using monitors, but cannot be implemented using semaphores.

Fill in the answers for the true/false questions into this table

1		**9**	
2		**10**	
3		**11**	
4		**12**	
5		**13**	
6		**14**	
7		**15**	
8			

Name ______________________________ ID# ____________________

PART B

Question 1:

State the four conditions that need to hold simultaneously for deadlock to occur?

Question 2:

Give one advantage and one disadvantage of a large scheduling quantum?

Question 3:

Test-and-Set. The Test-and-Set instruction is used in hardware to achieve synchronization. It can be defined in the following way:

```
function Test-and-Set (var target: boolean): boolean;
    begin
        Test-and-Set := target;
        target := true;
    end;
```

Now show how the Test-and-Set instruction can be used to protect a critical region and hence achieve mutual exclusion (do not worry about satisfying the bounded waiting condition). Assume the Boolean variable used is "lock" and it is initialized to false

```
repeat



until false;
```

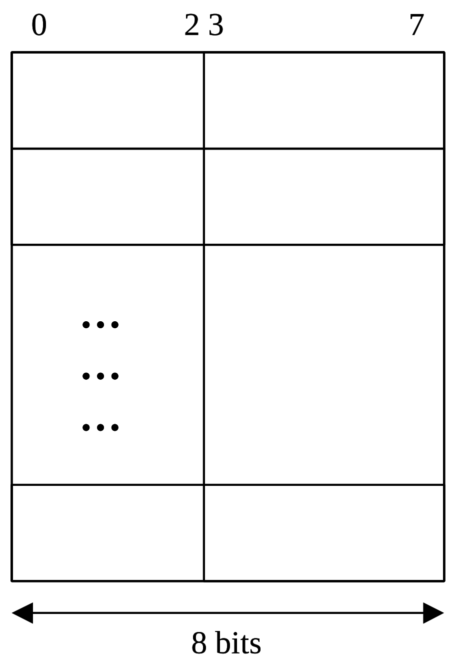
Name ______________________________________ ID# ____________________

Question 4:

A system runs at 100Hz. It takes 2 msec (including context switch overhead) to execute the clock interrupt handler. What fraction of CPU time is spent in servicing the clock?

Question 5:

Consider the interrupt table given below



Bits 0 through 2 are used to identify a unique ID for each interrupt.
Bits 3 through 7 are the function pointers, pointing to the interrupt service routine

a) What is the maximum number of unique interrupts Ids that can be assigned to devices using the above scheme?

b) What is the possible range of address space for interrupt service routines?

Name ____ ID# ____________________

PART C

Question 1:

Consider the following programs where **A** and **B** are arbitrary computations.

Initially, s1 = s2 = d =1; c1 = c2 = 0

P(s1)
c1 := c1 + 1
If (c1 == 1) then P(d)
V(s1)
A
P(s1)
c1 := c1 – 1
If (c1 == 0) then V(d)
V(s1)
	P(s2)
	c2 := c2 + 1
	If (c2 == 1) then P(d)
	V(s2)
	B
	P(s2)
	c2 := c2 – 1
	If (c2 == 0) then V(d)
	V(s2)

a) How many invocations of A can proceed concurrently?(Zero, one, two, any number) _____

b) While A is running how many invocations of B can proceed concurrently? (Zero, one, two, any number) _____

c) Can A starve? (Y/N) ______

d) Can B starve? (Y/N) _____

e) Is deadlock possible? (Y/N) ______

Name ______________________________ ID# ____________________

Question 2:

Consider the following snapshot of a system. P = Processes, R = resources

	MAX				ALLOCATION					
	P1	P2	P3	P4	P1	P2	P3	P4	Avail	Total
R1	3	6	3	4	1	6	2	0	0	9
R2	2	1	1	2	0	1	1	0	1	3
R3	2	3	4	2	0	2	1	2	1	6

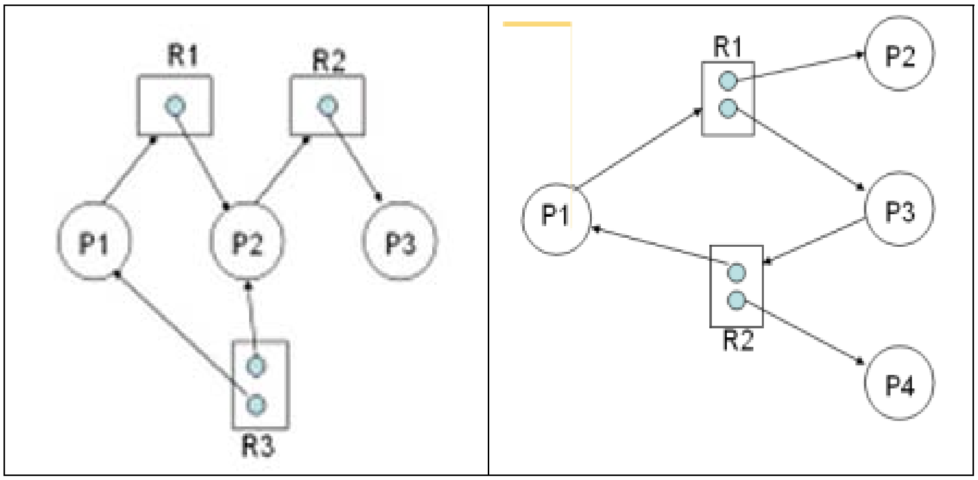
a) What is the content of the matrix Need? (Present the Matrix below)

b) Is the system in a safe state? If yes, give a safe sequence of processes. If your answer is no, explain why.

Name ______________________________ ID# ________________

Question 3:

Consider the following resource allocation graphs. [**5 points**]



(a) (b)

For each of the above graphs, identify whether the state is deadlocked? If so, then which of the processes are deadlocked?

a)

b)

Name ________________________________ ID# ____________________

Question 4:

The following is a suggested solution to the dinning philosophers problem. Analyze the following section of the code and answer the questions that follow:

```
VAR mutex: semaphore; /*binary semaphore*/
philosophers: array[max_philodophers] of semaphores; /*binary semaphores*/
chopsticks: array[max_philosophers] of boolean;
counter = 0; /*keeps track of how many philosophers have eaten during a cycle*/

procedure philosopher(i);
var right,left:integer;
begin
       wait(philosophers[i]);
       wait(mutex);
       counter := counter + 1;
       right := get_right_chopstick(i);
       left := get_left_chopstick(i);
       eat();
       if (counter = max_philosophers)
              while(counter>0)
              begin
                     signal(philosophers[counter]);
                     counter := counter -  1;
              end;
       signal(mutex);
end;
```

Answer the following questions in no more than 2 sentences:

a) Does the solution prevent deadlocks?

b) Does the solution prevent starvation?

c) How many philosophers can simultaneously eat?

Name ______________________________ ID# ____________________

PART C

Question 1:

Consider a logical address space of 128 pages of 1024 words each, mapped onto a physical memory of 32 frames.

a. How many bits are there in the logical address?

b. How many bits are there in the physical address?

Question 2:

Consider a paging system with the page table stored in memory.

a. If memory reference takes 200ns, how long does a paged memory reference take?

b. If we add associative registers and 75 percent of all the page-table references are found in the associative registers, what is the effective memory reference time? (Assuming that finding a page-table entry in the associative registers take zero time, if the entry is there.)

Name ______________________________ ID# ________________

Question 3:

A pure paging system uses a 3-level page table. Virtual addresses are decomposed into 4 fields, **a**, **b**, **c** and **d**, with '**d**' being the offset. In terms of these constants, what is the maximum number of pages in the virtual address space?

Question 4:

A small virtual memory system has only 2 frames of physical memory, and uses demand paging. Both frames are initially empty. Consider the following page reference string:

1 2 3 2 3 1 3 1 2 1

Give the <u>lower bound</u> on the number of page faults that will result from this reference string. Write <u>one sentence</u> on how you arrived at your answer (<u>important</u>).

Name ______________________________ ID# ________________

PART D

Question 1:

Consider the following sequence of virtual memory references generated by a single program in a pure paging system:

10,11,104,170,73,309,185,245,246,434,458,364

(a) Derive the corresponding reference string assuming a page size of 100 words.

(b) Determine the number of page faults for each of the following page replacement strategies, assuming that two page frames are available to the program. Use the reference string you derived in part (a).

- **FIFO**
- **CLOCK**
- **LRU**

Name ________________________________ ID# ____________________

Question 2:

Consider the following set of processes:

Process ID	Arrival Time	Service Time
A	0	120
B	1	40
C	2	150
D	3	20

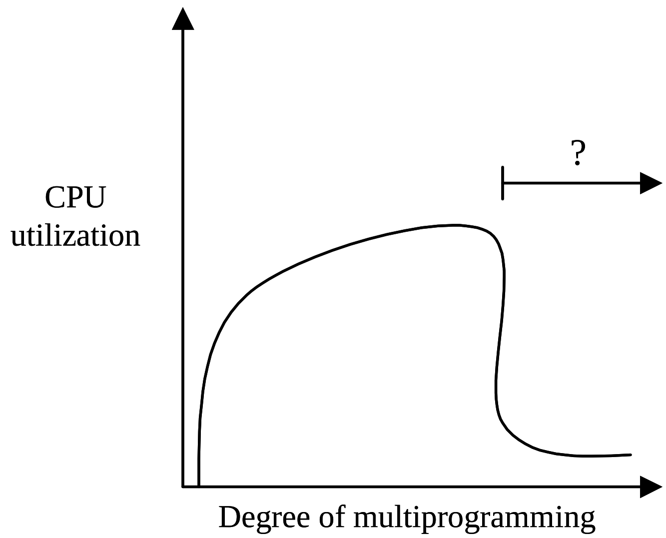
Calculate the Finish time, Waiting time and turnaround time for each of the above processes using the **First Come First Served** algorithm. You are expected to know what Finish, Turnaround and Waiting times mean, the proctors will NOT help you with that.

Scheduling Algorithm	Parameter	Process ID			
		A	B	C	D
	Finish Time				
First Come First Served	Turnaround Time				
	Waiting Time				

Name ________________________________ ID# ____________________

Question 3:

Consider the graph given below that shows a plot of “degree of multiprogramming” on **X**-axis versus “CPU utilization” on **Y**-axis.



a) What is the phenomenon observed by a sudden dip in the graph (as shown with the question mark) called?

b) How can it be prevented?