Where an algorithm is asked for, you may write in any suitable pseudocode. Correct syntax for any computer language is not expected.

Answer 3 questions.

1.

a) What three criteria must be satisfied by any solution to the critical section problem, and why? [6]

b) In the early UNIX operating systems there were no explicit semaphores. Instead, the following two system calls were provided:

i) `int creat(file_name)`
returns `0` if a file was created successfully, or `-1` if the file already exists.

ii) `unlink(file_name)`
deletes a file.

Assuming that these operations are atomic, use these primitives to implement general semaphores. You should end up with a series of functions which permit you to write lines like:

```
semaphore("s", 1)      /* Initialise semaphore to 1 */
P("s")
V("s")
```

State any assumptions you make and identify any limitations in your solution. [12]

c) The readers/writers problem has a variant in which:

i) New readers are delayed if a writer is waiting

ii) A delayed reader is awakened only if no writer is waiting

Using any concurrency control mechanism you wish, present and justify pseudocode which implements a solution to this problem. Of the criteria you presented in answer to section a), which does your solution address successfully. [15]

[TURN OVER]

2.

a) What conditions are necessary if deadlock is to occur? [4]

b) What is meant by an *unsafe state*? [2]

c) Consider a system which consists of four instances of a resource, shared by three processes, each of which needs at most two instances. Show that the system is deadlock free. [4]

d) How does *deadlock avoidance* differ from *deadlock detection*? What considerations would affect your choice between avoidance and detection, and what are the advantages and disadvantages of each? [12]

e) Explain the Banker's algorithm for deadlock avoidance. Use it to determine whether the following system is safe or not:

- A system has 5 processes, and the following total number of resources: 8 of $resource_A$, 8 of $resource_B$, 7 of $resource_C$, 3 of $resource_D$.
- The current state of the system is given by the following matrices:
  - Max:

	$resource_A$	$resource_B$	$resource_C$	$resource_D$
P_0	5	4	0	1
P_1	1	2	1	0
P_2	4	4	6	2
P_3	6	1	2	2
P_4	5	6	4	2

  - Need:

	$resource_A$	$resource_B$	$resource_C$	$resource_D$
P_0	1	2	0	0
P_1	0	1	1	0
P_2	3	2	4	1
P_3	5	1	0	1
P_4	4	4	3	2

[7]

f) Using the above system, would the following requests be permitted? Justify.

i) Process P_4 requests an extra 1 of $resource_A$ and 2 of $resource_B$

ii) Process P_2 requests an extra 2 of $resource_C$ [4]

[CONTINUED]

3.

a) What are the costs and benefits of implementing virtual memory? Can the costs exceed the benefits and, if so, what action can be taken to prevent this? [8]

b) When do page faults occur, and what actions does the operating system take when they occur? [4]

c) Why are page sizes always powers of 2? [2]

d) What is the difference between internal and external fragmentation? [4]

e) Devise and explain two segment-replacement algorithms based on the FIFO and LRU page-replacement schemes. Remember that segment sizes are not all the same, so the segment chosen for replacement may not be large enough to leave sufficient consecutive locations for the new segment. [15]

4.

a) What is the difference between one-way and two-way encryption? Why is one-way encryption used for maintaining UNIX passwords? [6]

b) If a salesman uses his laptop to access his customer files and his email remotely as he travels around the country, what threats are there to the security and integrity of the data which he is accessing? How can these threats be countered and how can he and the host computer which he is accessing be sure of each other's identity? [10]

c) Describe in detail the Needham Schroeder authentication procedure and the reason why it is necessary. For each step in the protocol, explain what is happening and why. [12]

d) *The US government are right to try to restrict the export of strong encryption on the grounds of national security.* In 150 words or fewer, argue for or against this proposition. [5]

[TURN OVER]

5.

a) What are meant by the terms *scheduler, turnaround time*, *i/o bound*, and *ageing*. [4]

b) Describe the *shortest job first* scheduling algorithm and show that it is optimal for non-preemptive systems. How would you actually implement such an algorithm? [6]

c) What is the difference between preemptive and non-preemptive algorithms? Why would a strict non-preemptive approach be unlikely to find favour with users of shared workstations? [4]

d) Explain the degree to which the following scheduling algorithms discriminate in favour of short processes:

i) First come first served
ii) Round robin
iii) Multilevel feedback queues [9]

e) In the following scenario, produce Gantt charts which illustrate the execution of processes, and from these derive the turnaround and wait times for each process for:

i) Shortest job first

ii) Shortest remaining time first

In both these cases, the context switch time is 2 ms.

Process	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**
Arrival time /ms	**1**	**2**	**5**	**10**	**13**	**16**	**18**
CPU burst /ms	**3**	**20**	**7**	**2**	**10**	**30**	**4**

State any assumptions you make. [10]

[END OF PAPER]