Question 1. *(10 points) Recall that Banker's algorithm is designed to avoid deadlock by only allocating resources to threads such that the current allocation remains in a 'safe' state. Explain what is meant by a safe state. If the system moves into an unsafe state, will deadlock occur?*

Question 2. *(10 points) Describe the difference between internal and external fragmentation. What effect does paging have on each type of fragmentation?*

Question 3. *(10 points) Explain why systems using paging usually choose a page size that is a power of 2 (e.g.,* $2^8 = 256$ *bytes,* $2^9 = 512$ *bytes, etc). What is the disadvantage of choosing a page size that is not a power of 2?*

Question 4. *(30 points) Consider a system with 32 byte pages and a total memory size of 2048 bytes. Assume that the system can access individual 4-byte words as the smallest unit of memory addressing (i.e., as in a typical 32-bit system). Recall that this means that an offset of 1 into a page means the second 4-byte word of the page (i.e., bytes 4-7).*

1. *(5 points) What is the total number of addressable words supported by this memory? How many different pages can be supported?*

2. *(5 points) How many bits are needed for an address? Of these, how many bits are needed for the page number* (p) *and how many for the offset* (d)*?*

3. *(15 points) Assuming the (partial) page table shown below, translate virtual address "28" to a physical address (i.e., the* k*th word of physical memory). To receive partial credit, be sure to show your work.*

Page	Frame
0	5
1	14
2	9
3	7
4	18
...	...

4. *(5 points) Suppose you extend your paging system to support segmented paging, where each process will have 7 segments. How many bits will be needed to encode a virtual address? How many bits for a physical address?*

Question 5. *(10 points) Explain an advantage and disadvantage to storing the page table in hardware versus in memory. Are these considerations any different when deciding how to store the segment table?*

Question 6. *(10 points) Suppose you want to run a set of applications which require more memory than your system has. Briefly explain two memory management techniques that your OS could use to support running these applications simultaneously.*

Question 7. *(20 points) Determine how the FIFO and MIN page replacement algorithms would handle the following page access pattern:* $A, B, C, D, E, A, B, E, D, B, B, A$. *As in the figures below, assume that the system has three frames of memory (each which can hold a single virtual page). Fill in the frame contents for each step of the access pattern and report the total number of page faults for each algorithm.*

FIFO	A	B	C	D	E	A	B	E	D	B	B	A
F1												
F2												
F3												
Fault?												

MIN	A	B	C	D	E	A	B	E	D	B	B	A
F1												
F2												
F3												
Fault?												