Assume a computer system that uses paging, where the size of the physical memory is 100 frames and each frame (page) is exactly 512 bytes large. Now assume that the five processes shown in the table below are loaded entirely into physical memory. How many frames are used by each process and what is the memory waste due to the internal/external fragmentation?

Process	Size (bytes)	# frames	Internal Fragmentation	External Fragmentation
A	1024			
B	513			
C	999			
D	2000			
E	480			

Assume the original design of a TLB has 20 entries and a page size of 2048 bytes. If the number of entries is doubled and the page size is cut in half, what is the new TLB reach?

In a 64 bit machine, with 256 MB RAM (2^{28}) and a 4KB page size (2^{12}), how many entries will there be in the page table if it is inverted?

a) 2^{16}
b) 2^{22}
c) 2^{14}
d) Not possible to determine (explain)

If the total number of available frames is 60 and there are 2 processes, one of 100 pages and the other of 50 pages, then how much of the memory would be proportionally allocated to each of these processes?

a) Depends on the scheduler
b) 40 and 20 frames respectively
c) Memory is allocated equally for both
d) 10 and 5 frames respectively

Consider the segment table shown below; what are the physical addresses for the following logical addresses (the logical addresses are in the format *(s,o)* where s is the segment and o is the offset): **(0,430)**, **(1,10)**, **(1,11)**, **(2,500)**

Segment	Base Address	Segment Length
0	219	600
1	2300	14
2	30	100
3	1327	580
4	1952	9

Consider a system with five resources **< R0, R1, R2, R3, R4 >**, where the total number of each system resource is **< 10,8,5,3,6 >** (i.e., there are 10 instances of resource R0, 8 of R1, etc.). The two tables below show the current allocations of these resources to processes P1..P5 and the maximum number of additional resources needed by each process.

- Use the **banker's algorithm** to explain why the current state is safe or unsafe (show each step of your analysis).
- Would **a request by process P4 for 1 additional instance of R3** be granted or not (justify your answer clearly and show each step of your analysis)?
- Would **a request by process P1 for 1 additional instance of R2** be granted or not?
- Would a **request by process P2 for 1 additional instance of R1** be granted or not? (Again, justify your answer clearly for each question.)

Current Allocations

Process	R0	R1	R2	R3	R4
P1	2	1	1	1	1
P2	4	3	0	0	0
P3	1	1	2	0	1
P4	2	0	1	0	1
P5	0	1	1	1	0

Max. Additional Resources

Process	R0	R1	R2	R3	R4
P1	3	2	1	1	0
P2	1	1	0	1	1
P3	1	0	2	1	2
P4	2	1	0	0	1
P5	1	1	2	2	0

In on-demand paging, a page replacement policy is used to manage system resources. Suppose that a newly created process has 3 frames available, and then generates the page references indicated below (the page table is initially empty).

	E	D	H	B	D	E	D	A	E	B	E	D	E	B	G
F1	E	E													
F2		D													
F3															
PF	X	X													

Suppose that primary memory is managed using a free-list to keep track of holes (spaces that have not been allocated) and that the **best-fit heuristic** is used for space allocation. At a certain time, the free-list looks as follows:

(100,500) → (700,50) → (1000,200) → (4000,700) → (4900,100)

where the first number indicates the **starting address** of the hole and the second number is the **size** of the hole, in bytes. Suppose that the following events occur in the order given:

- 250 bytes of space starting at address 1200 are freed.
- 400 bytes of space are allocated.
- 100 bytes of space starting at address 600 are freed.

Show the smallest possible free-list after these events have occurred (i.e., the list should have as few entries as possible). Also, when allocating space from a hole, always allocate memory from the beginning of a hole.