1. (2 pt.)

a.

Sol)

- Since Bit map size is 8KB = $8 \times 2^{10} \times 8$ (bit), there are 64×2^{10} units.
- Total RAM size = # of unit $\times$ size of unit = $(64 \times 2^{10}) \times (2 \times 2^{10}) = 128 \times 2^{20}$Byte = 128 MB

b.

Sol)

- The linked list: number of node for linked list= 128 MB /64KB = $2^{27} / 2^{16}$ or 2^{11} nodes.
- size of each node = 32+16+16 = 64 bit = 8 byte = 2^3 bytes
- Total size of linked list = number of node $\times$ size of a node = $2^{11} \times 2^3$ bytes = **2^{14} bytes**.

2. (1 pt.)

- First Fit 9(21) -10(12) -15(20)- 18(18)
- Best Fit 9(14) – 10(15)-15(18)-18(20)
- Worst Fit 9 (35) – 10(26) -15 (25) – 18 (23)
- Next Fit 9(21)-10(20)-15(18) -18(25) or
  - 9(21) -10(12)- 15(20) -18(18)

3. (2 pt.)

a) FIFO : 15 page fault

5,	0,	1,	2,	0,	3,	0,	4,	2,	3,	0,	3,	2,	1,	2,	0,	1,	5,	0,	1
5	0	1	2		3	0	4	2	3	0			1	2			5	0	1
	5	0	1		2	3	0	4	2	3			0	1			2	5	0
		5	0		1	2	3	0	4	2			3	0			1	2	5

b) LRU: 12 page fault

5,	0,	1,	2,	0,	3,	0,	4,	2,	3,	0,	3,	2,	1,	2,	0,	1,	5,	0,	1
5	0	1	2		2		4	4	4	0			1		1		1		
	5	0	1		3		3	2	2	2			2		2		5		
		5	0		0		0	0	3	3			3		0		0		

c) Optimal: 9 page fault

5,	0,	1,	2,	0,	3,	0,	4,	2,	3,	0,	3,	2,	1,	2,	0,	1,	5,	0,	1
5	0	1	1		3		3			3			1				1		
	5	0	0		0		4			0			0				0		
		5	2		2		2			2			2				5		

4. (2 pt.)

a.

Sol) # of page entries = size of page table / entry size = 8MB / (64/8) = 2^{20} entries
Page size = possible virtual space / # entries = 2^{32} /2^{20} = 4×2^{10} = 4 KB

b.

Sol) # of page frame = 2^{21} page frames
Size of RAM = size of page frame × number of page frames
= 4KB × 2^{21} = $4 \times 2^{10} \times 2^{21} = 8 \times 2^{30}$ = 8 GB

5. (1 pt.)

- Long-Term Scheduler – Selects a process from the pool of job and load into memory for execution
- Short-term scheduler – selects a process from the ready queue and allocates the CPU.
- Memory Scheduler – schedule which process is in memory and in the secondary memory of swap area.

6. (1 pt.)

Sol) To access an instruction located in virtual address I, OS need two memory access times.

- Access a memory to get page frame number from page table in the memory
- Now calculate physical address by combining page frame number + offset

7. (1 pt.)

a.

$$Total\ Overhead(P) = Average\ page\ table\ size + the\ wasted\ memory\ in\ th\ last\ page\ of\ process$$
$$= \frac{S}{P} \times E + \frac{P}{2}$$

b.

$$Overhead'(P) = -\frac{SE}{P^2} + \frac{1}{2} = 0$$

$P = \sqrt{2SE}$: *optimal* page size