1. (10 pt.)
   a)

$$E = [3\ \ 3\ \ 4\ \ 3],\ C = \begin{bmatrix} 1010 \\ 0101 \\ 1100 \\ 0011 \\ 1001 \\ 0110 \end{bmatrix},\ \ R = \begin{bmatrix} 0100 \\ 1000 \\ 0010 \\ 0100 \\ 0010 \\ 1000 \end{bmatrix},\ A = [0010]$$

$$\text{Sol) } A = [0010] \xrightarrow{P_3} [1110] \xrightarrow{P_1} [2120] \xrightarrow{P_2} [2221] \xrightarrow{P_4} [2232] \xrightarrow{P_5} [3243] \xrightarrow{P_6} [3343]$$

   b)

$$E = [3\ \ 2\ \ 3],\ C = \begin{bmatrix} 101 \\ 110 \\ 011 \\ 100 \\ 001 \end{bmatrix}, R = \begin{bmatrix} 010 \\ 101 \\ 100 \\ 010 \\ 100 \end{bmatrix}$$

Sol) A = (000)

2. (5 pt.)
   a. What are four necessary conditions for a deadlock
      1. Mutual exclusion
      2. Hold-and Wait
      3. No preemption
      4. Circular wait
   b. Four strategies for dealing with a deadlock
      1. Ignore
      2. Detection and recovery
      3. Avoidance with dynamic allocation
      4. By attacking one of necessary deadlock condition
   c. When OS detect a deadlock, what are three possible ways to recover from the deadlock?
      1. preemption
      2. roll back
      3. terminate a process one by one until break from the deadlock
   d. Why is deadlock avoidance impractical?

Deadlock avoidance requires future knowledge of resource requests, meaning the OS must know in advance

e. Is it true deadlock prevention guarantees no deadlock?

Yes since it break one of four necessary conditions

3. (5 pt.)

Solution 1)

Sol)
Attacking hold and wait,
starvation

Solution 2)

-

Sol)
Attacking circular wait,
If a process need two resource at a same time, this solution have problem

4. (10 pt.)

Process	Allocated			Max Need			R		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	7	4	3
P_1	2	0	0	3	2	2	1	2	2
P_2	3	0	2	9	0	2	6	0	0
P_3	2	1	1	2	2	2	0	1	1
P_4	0	0	2	4	3	3	4	3	1

A = (3, 3, 2)

a) Will a request of (1, 0, 2) by P_1 be granted? (it is not yes/no problem)

Process	C			R			A		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	4	3	2	3	0
P_1	3	0	2	0	2	0			
P_2	3	0	2	6	0	0			
P_3	2	1	1	0	1	1			
P_4	0	0	2	4	3	1			

A = (2, 3, 0) – P_1 – (5, 3, 2) – P_3 – (7, 4, 3) –P_0 –(7, 5, 3)- P_2 –(10, 5, 5) –P_4 –(10, 5, 7)

b) Will a request of (3, 2, 0) by P_4 be granted? (it is not yes/no problem)

Process	C			R			A		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	4	3	0	1	2
P_1	2	0	0	1	2	2			
P_2	3	0	2	6	0	0			
P_3	2	1	1	0	1	1			
P_4	3	2	2	1	1	1			

A = (0, 1, 2) – P_3 – (2, 2, 3) – P_1 – (4, 2, 3) –P_4 –(7, 4, 5)- P_0 –(7, 4, 5) –P_2 –(10, 5, 7)

c) Will a request of (3, 3, 0) by P_4 be granted? (it is not yes/no problem)

Process	C			R			A		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	2	3	0	0	2
P_1	2	0	0	1	2	2			
P_2	3	0	2	6	0	0			
P_3	2	1	1	0	1	1			
P_4	3	3	2	1	0	1			

Answer) non- of process can

5. (10 pt.)

	Allocated					Need More					Available				
	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5
P_1	1	0	2	1	1	0	1	0	0	2	0	0	X	1	Y
P_2	2	0	1	1	0	0	2	1	0	0					
P_3	1	1	0	1	0	1	0	3	0	0					
P_4	1	1	1	1	0	0	0	1	1	1					

What is the minimum value of X and Y required for this to be considered a safe state? To ensure the accuracy of your answer, you should describe the logical process by which the values of X and Y are selected.

Sol) since P_1 need 3 R_5 in total minimum Y should be >=2.
since R_1=0, R_2 =0, only P_4 can be selected based on A with X >=1

- with X=1, Y=2 A=(0 0 1 1 2)
  after P_4, A = (0 0 1 1 2) + (1 1 1 1 0) = (1 1 2 2 2)
  after P_1, A = (1 1 2 2 2) + (1 0 2 1 1) = (2 1 4 3 3)
  after P_3, A = (2 1 4 3 3) + (1 1 0 1 0) = (3 2 4 4 3)
  after P2, A = (3 2 4 4 3) + (2 0 1 1 0) = (5, 2 5 5 3)

6. (10 pt.)
   a. Which deadlock condition is removed? Hold and wait
   b. What is the main disadvantage? Poor resource utilization. Starvation risk, process wait too long
   c. Give real-world inefficiency example. Process P1 hold two resources R1 and R2 and currently only use R1. Process P2 only need R2 but need wait.