Set C Part 1

Design Marks: Total = 7

read(A,B, n)

1. read the value of *n*
2. **for** *i* ← 1 to *n*
   do read *emp_id*
   read *salary*
   A[i] ← *emp_id*
   B[i] ← *salary*

Evaluation criteria : **[1 mark]** Reading the values emp_id and salary and storing it into arrays A and B respectively.

Find_Position(A,B,n,e)

// using two extra arrays

1. **for** i ← 0 to n-1
   do if A[i] = e
   then sal ← B[i]
   pos ← i
2. Create two arrays D and E of size *n*
3. Initialize j ← 0
4. **for** i ← 0 to n-1
   do if B[i] < sal
   then D[j] ← A[i]
   E[j] ← B[i]
   j ← j+1
5. D[j] ← e
6. E[j] ← sal
7. j ← j+1
8. final_pos ← j
9. **for** i ← 0 to n-1

do if B[i] > sal

then D[j] ← A[i]

E[j] ← B[i]

j ← j+1

10. **for** i ← 0 to n-1

do A[i] ← D[i]

B[i] ← E[i]

11. Print *final_pos* and the two arrays A and B

Evaluation criteria : **[6 marks]**	
Division:	Finding the sal of e - 1 mark
	Finding the final position of e - 2 marks
	Preserving the relative positions of elements - 3 marks

Set C Part 2

Design Marks: Total = 3

read(A,B, n)

// read employee details to arrays A and B

1. read the value of *n*
2. **for** *i* ← 0 to *n*-1

do read *emp_id*

read *salary*

A[i] ← *emp_id*

B[i] ← *salary*

3. read the value of *k*

Find_Highest(A, B, l, r, k) *// initially l= 0, r = n-1*

// Slightly modify *Find_Position(A, B, n, e)* to *Find_Position (A, B, l, r, e)* that positions *e* between *l* and r (inclusive) and returns the value *final_pos* (1 <= *final_pos* <= *r-l+1*) of the employee *e.*

1. $e \leftarrow A[l]$ // Choose an arbitrary employee e in array A.

   // Here we take the leftmost employee

2. $n \leftarrow r - l + 1$
3. $pos \leftarrow Find\_Position\ (A, B, l, r, e)$
4. $index = pos - 1$
5. **if** $index = n - k$ // After positioning, k^{th} highest is in the $(n - k)^{th}$ index

   then return A[*index*]

6. **else if** $index > n - k$

   then return $Find\_Highest(A, B, l, index - 1, k - (r - index))$

7. **else return** $Find\_Highest(A, B, index + 1, r , k)$

Evaluation criteria : **[3 marks]**	
Division:	Modification of *Find_Position* ( ) - 1 mark
	Proper recursion calls of *Find_Highest*( )- 2 marks