Q1:

A) define the meaning of an Algorithm and it's Correctness.

B) contract the problem complexity and algorithm complexity.

C) For the following pair of functions, determine the smallest integer value of n≥0 for which the first function becomed greater than or equal to the second function.

- n^2 , 10n
- 2^n , $2n^3$

D) determine the total number of iteration in the following algorithm:

```
i=1
Loop (i<= 10)
        j=1
        Loop (j<=10)
                Application code
                        j= j * 2
        i= i+1
```

Q2: write the pesudo code for the Insertion Sort and prove that it's worst case time is O(n2).

Q1:

a) Algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.

Algorithm Correctness : An algorithm is correct if it works correctly for all legitimate inputs (all instances). An input to an algorithm specifies an instance of the problem the algorithm solves.

b) • Problem Complexity

-Can the problem be solved?

-How difficult is the problem? Does there exist a better algorithm?

• **Algorithm Complexity**

Time complexity

- **How much time it takes to compute the output measured as a function T(N) of the input Size N.**

Space complexity

- **How much memory it takes to compute the output, measured as a function S(N) of the input size N. Alternatively, How much is the peak memory during computation.**

c)

n^2 , 10n

n	n^2	10n
0	0	0
1	1	10
2	4	20
3	9	30
4	16	40
5	25	50
6	36	60
7	49	70
8	64	80
9	81	90
10	100	100
11	121	110
12	144	120

2^n , $2n^3$

n	2^n	$2n^3$
0	1	0
1	2	2
2	4	16
3	8	54
4	16	128
5	32	250
6	64	432
7	128	686
8	265	1024
9	512	1458
10	1024	2000
11	2048	2662
12	4096	3456
13	8192	4394
14	16384	5488

d) #Iterations = #outer loop iterations x #inner loop iterations

The number of iterations in the inner loop is log2 10 . In the above program code, the inner loop is controlled by an outer loop. The above formula must be multiplied by the number of times the outer loop executes, which is 10. this gives us 10 x log2 10

In general, T(N) = N x log2 N

Q2:

Insertion Sort: Pseudo Code

The time "cost" of each statement and the number of times each statement is executed.
For each j let tj denote the number of times the **while** loop test in line 5 is executed.

Insertion sort (A,n)	Cost	times
for j←2 to n do	C_1	n
key ← A[j]	C_2	n-1
// insert A[j] into the sorted sequence		
i ← j -1	C_4	n-1
while i > 0 and A[i] > key do	C_5	$\sum_{j=2}^{n} tj$
A[i+1] ← A[i]	C_6	$\sum_{j=2}^{n} tj-1$
i ← i -1	C_7	$\sum_{j=2}^{n} tj-1$
A[i+1] ← key	C_8	n-1

Insertion Sort: Running Time

Worst case: The longest running time of any input of size n.
The worst case of Insertion Sort occurs if the array is in the reverse order. Thus tj = j for j = 2, 3, …, n

Since $$\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1$$

$$\begin{aligned} T(n) &= c_1 n + c_2(n-1) + c_4(n-1) + c_5\left(\frac{n(n+1)}{2} - 1\right) \\ &\quad + c_6\left(\frac{n(n-1)}{2}\right) + c_7\left(\frac{n(n-1)}{2}\right) + c_8(n-1) \\ &= \left(\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2}\right)n^2 + \left(c_1 + c_2 + c_4 + \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} + c_8\right)n \\ &\quad - (c_2 + c_4 + c_5 + c_8)\ . \end{aligned}$$

T(n) can expressed as a quadratic Function *T(n)* = $an^2 + bn + c$
where a, b, and c are constants that depend on the statement costs ci