## **Question 1:**

# Find the running time as a function T(*n*) of input size *n* in each of the following cases considering only the number of iterations of the loops (only the number of comparison operations in the loops)

# Express the growth rate of the function in Big O notation.

**(a) s = 0;**

**for (i = n; i > = 1; i--)**

**s = s+1;**

# s = 0;

# for (i = 1; i < = n; i++)

**for (j = i; j < = n; j++)**

**s= s+1;**

**(c ) float power( float a, int n)**

**{**

**float p = a;**

**float r = 1;**

**int m = n;**

**while n > 0 do**

**if n is odd**

**n--;**

**r = r · p**

**else**

**n = n/2;**

**p = p.p;**

**end if**

**end while**

**return r}**

## **Question 2:**

* Express the following functions in terms of Big-O notation (a, b and c are constants). Which one is the Best? Which one is the worst?

1. f(n) = an2 + bn + c
2. f(n) = 2n + n log n + c
3. f(n) = n log n+ b log n + c
4. 3(n+1)7+2n log n