Question 1: Calculate running time of algorithm.

for(i = 1; i <= n\*n; i=i+1)

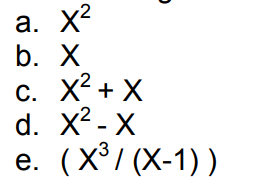
for (j = 1; j <= i; j++)

print(“Hello”);

Question 2: Write an algorithm to sum the all integers in the given array. Calculate the running time of the algorithm as well.

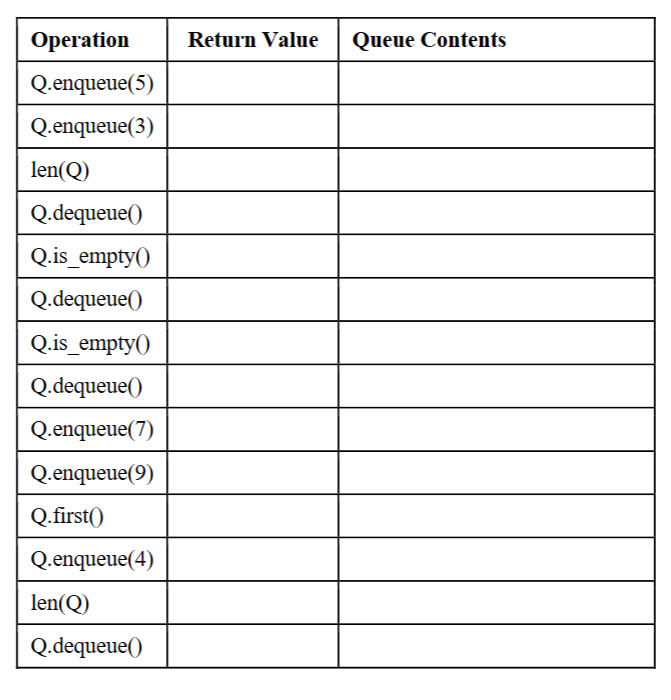
Question 3 : What will the Time Complexity of stack Push and pop operations when it is implemented on a Linked List? Compare it with array based stack in terms of performance.

Question 4: Group the following into equivalent Big-Oh functions:



Question 5: What are the Constant Factors in any algorithm? How is growth rate affected by the constant factors? How do you handle constant factors in Big-Oh

Question 6: Consider a Queue Perform the Operations mentioned in the table below.



Question 7: Convert the infix to postfix Expression show the state of stack and postfix expression at each step of conversion

( A + B \* C – D ) / ( E \* F )

Question 8: Table

Description automatically generated

Question 8: Consider the list of characters in your full name (all in upper case letters). Using Linear Search, design an algorithm   
that counts the number of first letter of your first name in your last name.   
   
For example, given complete name is:   
• STEVE ROGERS   
The first letter of the given first name is “S”. Using Linear Search, the algorithm should look for “S” in the given last   
name, i.e. “ROGERS”, and calculate how many times it has appeared in it. In this specific example, the answer will   
be 1 as “S” appears only once in “ROGERS”   
   
Please show the output based on your name, for which you must state your complete name so it can be compared for the final result. Use Linear Search algorithm Assume name is taken as input with only first and last names (no middle name or extra family name). Irrespective of the input, all the letters should be in upper case.

**Question 9:**

Consider the list of characters in your full name (all in upper case letters). Search for the middle letter of your first name in your last name using Binary Search, and return the index at which it was found. Handle the case when a character cannot be found in the list.   
   
For example, given complete name is:   
• STEVE ROGERS   
The middle letter of the given first name is “E”. Using Binary Search, the algorithm should look for “E” in the given   
last name, i.e. “ROGERS”, and return its index. In this specific example, the answer will be 0 as “E” appears at   
index 0 in the sorted last name, i.e. “EGORRS”   
   
Please show the output based on your name, for which you must state your complete name so it can be compared for the final result.   
If there are even number of letters in your first name, there will be two middle letters. In that case choose the first letter, i.e. the floor value If the middle letter appears multiple times in the last name, it should return the index of the first match Assume name is taken as input with only first and last names (no middle name or extra family name). Irrespective of the   
input, all the letters should be in upper case.

Question 10: Table

Description automatically generated

Question 11: Justify if the given time complexities hold true for the respective functions:   
a. 𝑓(𝑛)=5𝑛2 +3𝑛log𝑛+2𝑛+5   
𝑔(𝑛)= 𝑛2   
That is, time complexity = O(n2)   
 • ANSWER: If c = 15, n0 = 1, it will be justified  
 b. 𝑓(𝑛)=3log𝑛+2   
𝑔(𝑛)=𝑛log𝑛   
That is, time complexity = O(n log n)   
 ANSWER: If c = 5, f(n) <= c\*g(n) holds true.

What is n0?

For n0=1 , 3\*log(1)+2

0+2= 2 <= 5\*1log(1)

2<=0 false

For n0=2 , 3 \* log(2)+2 < = 5 \* 2 log (2)

3\*1+2 <= 5\*2(1)

5<=10

For n0=3, 3 \* log(3)+2 <= 5\* 3 log(3)

3\*1.585 +2 <= 5\*3(1.585)

6.755 < = 23.77

So we say that for c=5 and n0>=2 this time complexity is valid.

c. 𝑓(𝑛)=2𝑛+2   
𝑔(𝑛)=2𝑛   
That is, time complexity = O(2n)   
 ANSWER: If c = 4, n0 = 1, it will be justified  
   
 d. 𝑓(𝑛)=2𝑛+100log𝑛   
𝑔(𝑛)=√𝑛   
That is, time complexity = O(√𝑛)   
ANSWER: If c = 102, and n0 = 1, it will be justified. If c = 102, and n0 > 1, it will never hold true. Even if you change c, it won’t be justified, so time complexity is not O( 𝑛)