Ans1:  
  
A white paper with yellow writing

Description automatically generated

Upon analyzing, we can see that no matter how large the constant c is going to be, n8 will eventually surpass the c3 as n goes to infinity because the growth rate of n8 is much higher than a constant value c3 .  
  
  
Thus our contraduction leads towards a contradiction which further prooves that n3

answer 2:  
for i in range(n):

for j in range(n):

if( i == j):

The following condition exists inside the inner loop: ***if(i == j):***. This requirement can only be met if **i** and **j** are equal. The condition will be true **n** times in total (one for each iteration of the outer loop) since **i** and **j** both run from **0 to n-1.**  
  
Inside this condition, there is another loop:

for k in range(n\*n):

sum = i + j + k

Since the innermost loop runs ***n2*** times and is executed ***n*** times due to the condition, it gives n \* n2= n3 iterations.

Which makes the final answer:

***O(n3)***

Answer 3:  
  
***Loop Invariant:***  
At the end of each iteration ***i***, the variable ***answer*** holds the sum of first ***i + 1*** elements of list **A**.

***Initialization***:

At the end of the first iteration (***i = 0***), the variable **answer** holds the value of the first element ***A[0]***, therefore the loop invariant is true for ***i = 0***.

***Maintenance:***

Assume that the loop variant is true for some arbitrary iteration ***k***, then for iteration ***k + 1***, the value **A[k + 1]** gets added to the variable **answer**, therefore answer holds the sum of first **k + 2** (because array A is 0 indexed) elements of array **A**. Therefore the loop invariant holds true for iteration **k + 1**.

**Termination:**  
When the loop terminates, i equals len(A) - 1, meaning that answer holds the sum of all elements in the list `A`.

Answer 4:  
  
**Part 1:  
  
Assertion 1:**At the start of each iteration **i**, `temp` holds the value ***max(5, a[:i])***.

**Assertion 2:**After the loop exits, **temp = max(5, max(a))**

**------------------------------------------------------------------------------------------------------------------**

**Part 2:**lets take the base case where temp is initialized as 5 and as i = 0, no other value has been compared yet. **Inductive step:**lets assume that assertion holds true for some arbitrary index k, which means:  
temp=max(5, max(a[0],a[1],...,a[k−1]) )  
  
***Inductive hypothesis:***  
Lets prove that the above proves true for index of k+1. therefore, we can say that

*temp=*max(5, max(a[0],a[1],...,a[k−1]) )

*which means:*

*temp=max(5, max(a[0],a[1],...,a[k−1]) )*

**Now lets discuss in two separate cases,**

**Case 1: a[k] <= temp**

Here, the assertion retains it's true nature as the max value of temp should be temp and k is not greater than temp which means it will not change it.

**Case 2: a[k] > temp**

Temp takes the value of a[k] as it’s the new maximum. So, the maximum value up to index k becomes a[k], and the assertion holds.

**Conclusion:**

Hence we just proved that the assertion holds for every iteration i in the range of the length of the list a in the below:  
***temp=max(5, max(a[0],a[1],...,a[k−1]) )***

*Assertion 1 is supported by the inductive proof, which shows that the variable 'temp' always contains the maximum value of the items in the array 'a' that have been iterated over.*