

```
1. a.) for(i=1;i<=n;i++)  
        for(j=i;i<=n;j++)  
            printf("Hi");
```

Answer: -

for outer loop time complexity =  $O(n)$

for inner loop time complexity =  $O(n)$

for line3 time complexity =  $O(1)$

Then the final time complexity =  $O(n) * O(n) * O(1) = O(n^2)$

```
b.) for(i=1;i<=n;i*=3)  
        for(j=1;i<=n;j++)  
            printf("Hello");
```

Answer: -

Line 1 will run  $\log n$  times. The time complexity becomes  $O(\log_3 n)$  [i.e  $\log n$  base 3].

Line 2, the inner loop will run a max of  $n-1$  times for each outer loop iteration. Time complexity is  $O(n-1) = O(n)$

Line 3 is of constant complexity  $O(1)$ .

Total time complexity is  $O(\log n) * O(n) * O(1) = O(n \log 3 n)$