Question 4:
Prove that the Big-O of the following loop is O(N)

```
for (int i = 1; i <= n; i += c) {
  // some O(1) expressions
}
```

Ans:
Here c = Constant, means during the loop **I** will be increased by c.
The loop will be executed n times, means O(1) will be executed n time
So the Complexity of the loop Will be:= n * O(1)
= O(n*1)
= O(n)

Question 5:
Find the Big-O of the following loops // c is constant

```
for (int i = 1; i <=n; i += c) {
            for (int j = 1; j <=n; j = pow(i, c)) {
                        // some O(1) expressions
             }
}
for (int i = n; i > 0; i += c) {
            for (int j = i+1; j <= n; j *= c)
                        { // some O(1) expressions
            }
}
```

Ans:

```
for (int i = 1; i <=n; i += c) {
            for (int j = 1; j <=n; j = pow(i, c)) {
                        // some O(1) expressions
             }
}
```

O(loglogn)

O(n) * O(loglogn)

```
for (int i = n; i > 0; i += c) {
            for (int j = i+1; j <= n; j *= c)
                        { // some O(1) expressions
            }
}
```

O(logn)

O(n) * O(logn)

$= O(n) * O(loglogn) + O(n)* O(logn)$
$=O(n*loglogn) + O(n*logn)$