1) What is the time complexity of the following code snippet

for (int i = 1; i <= n; i += 2) {

System.out.print(i);

}

**O(n^3)**

**O(n^2)**

**O(n) correct Answer**

**O(logn)**

2) What is the time complexity of the following code snippet

static void solve(int N, int M) {

for (int i = 1; i &lt;= N; i++) {

if (N % i == 0)

System.out.println(i);

}

for (int i = 1; i &lt;= M; i++) {

if (M % i == 0)

System.out.println(i);

}

}

**O(N)**

**O(M)**

**O(N+M) correct Answer**

**O(NM)**

3) What is the time complexity of the following code :

static int func(int n) {

int s = 0;

for (int i = 1; i &lt;= 100; i++) {

s += i;

}

return s;

}

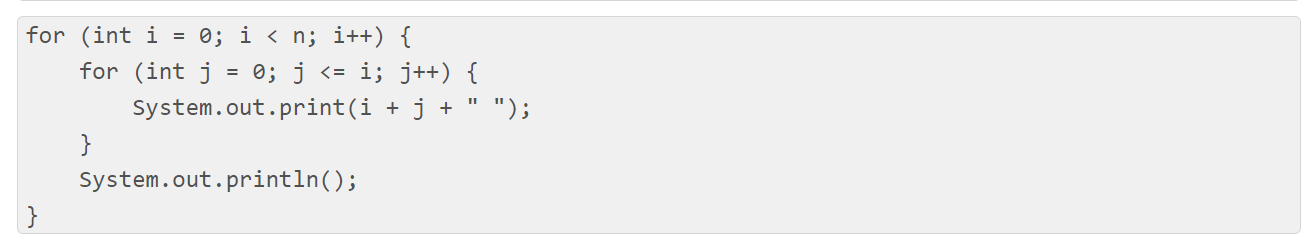
**O(n)**

**O(n^2)**

**O(s)**

**O(1)correct answer**

What is the time complexity of the following code :



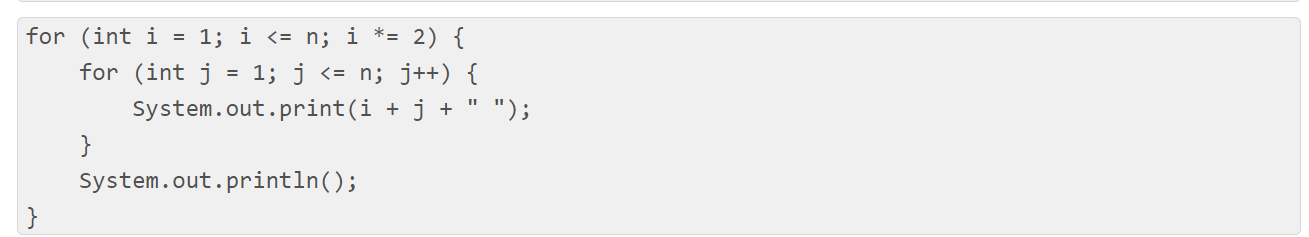
**O(n^2)correct answer**

**O(n)**

**O(nlogn)**

**O(n^3)**

What is the time complexity of the following code :



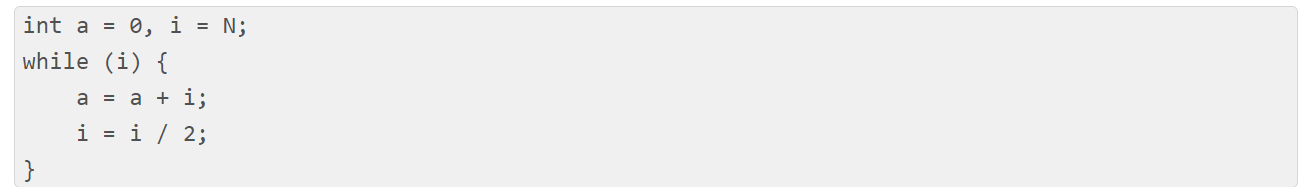
**O(n^2)**

**O(nlogn)correct answer**

**O(n)**

**O(1)**

What is the time complexity of the following code :



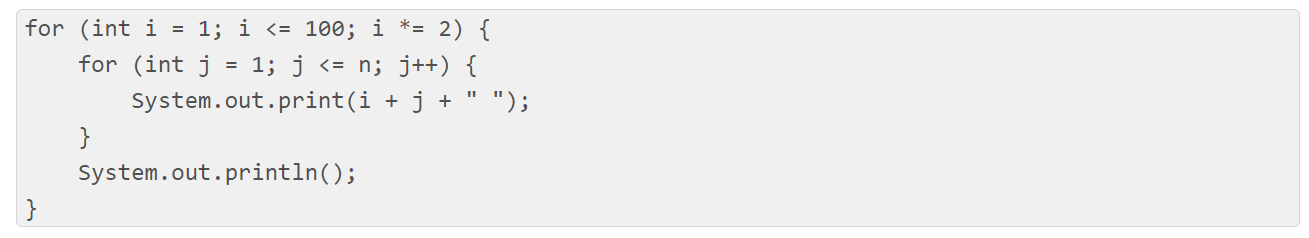
**O(N)**

**O(Sqrt(N))**

**O(N / 2)**

**O(log N)**

What is the time complexity of the following code :



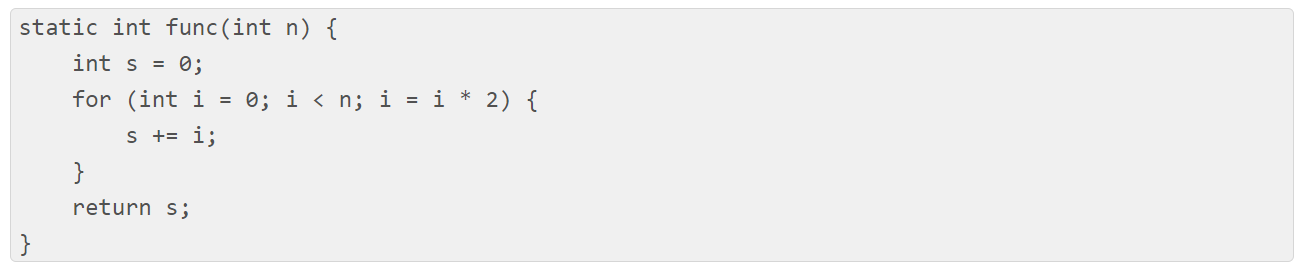
**O(n^2)**

**O(n)**

**0(nlogn)**

**O(1)**

What is the time complexity of the following code :



**O(n)**

**O(n^(1/2))**

**O(logn)**

**O(∞)**

What is the time complexity of the following code :



**O(N)**

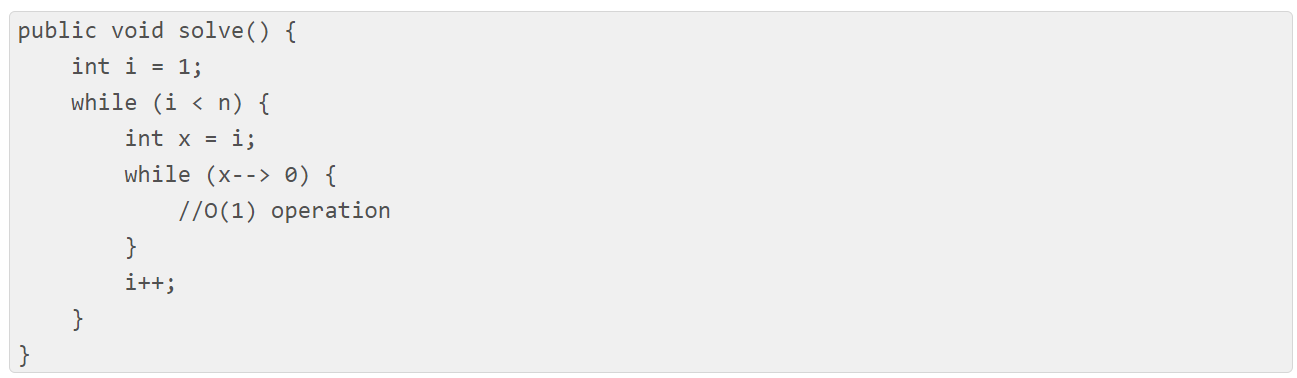
**O(N\*N)**

**O(Nlog(N))**

**O(log(N)) {Base 3}**

**O(log(N)) {Base 2}**

What is the time complexity of the following code :



**O(nlogn)**

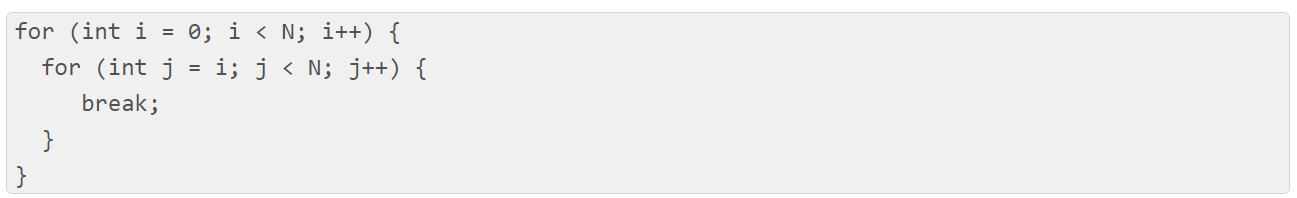
**O(n)**

**O(n sqrt(n))**

**O(n^2)**

**None of the above**

What is the time complexity of the following code :



**O(N\*N)**

**O(N\*log(N))**

**O(N)**

**O(log(N))**

**None of the above**

What is the time complexity of the following code :



**O(N)**

**O(N\*log(N))**

**O(N \* Sqrt(N))**

**O(N\*N)**

What is the time complexity of the following code :

int func(int n){

int s = 0;

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

s = s + i;

}

return s;

}

**O(n^(1/4))**

**O(n^(1/3))**

**O(n^(1/2))**

**O(n)**

What is the time complexity of the following code :



**O(nlogn)**

**O(n)**

**O(n^2)**

**O(1)**

**None of the above**

What is the time complexity of the following code :

for(int i = 1 ; i <= n ; i++){

for(int j = 1 ; j <= 3^i ; j++){

print(i + j);

}

}

**O(n^2)**

**O(nlogn)**

**O(2^n)**

**O(3^n)**

If an algorithm has a time complexity of O(1), then the complexity of it is ?

**constant**

**polynomial**

**exponential**

**none of the mentioned**

If for an algorithm time complexity is given by O(log2n) then complexity will:

**constant**

**polynomial**

**exponential**

**none of the mentioned**

If an algorithm has a time complexity of O(n), then the complexity of it is ?

**constant**

**linear**

**exponential**

**none of the mentioned**

If for an algorithm time complexity is given by O((3/2)^n) then complexity will:

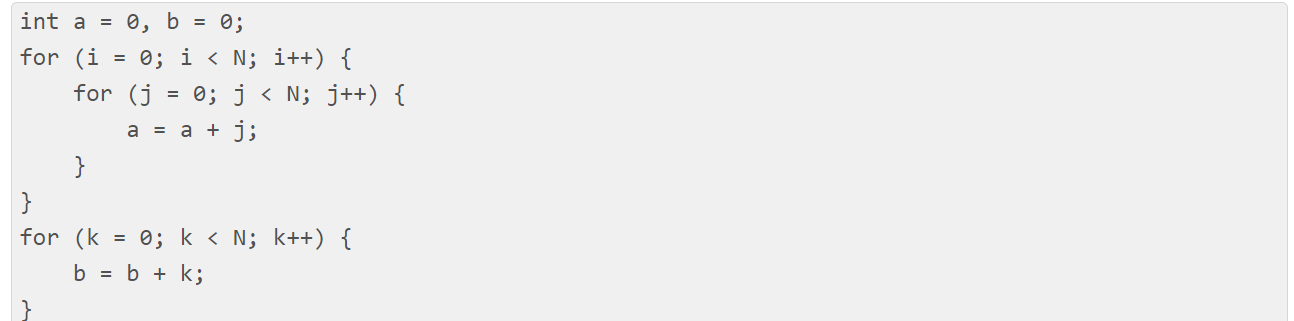
**constant**

**quardratic**

**exponential**

**none of the mentioned**

What is the time, space complexity of following code :



**O(N \* N) time, O(1) space**

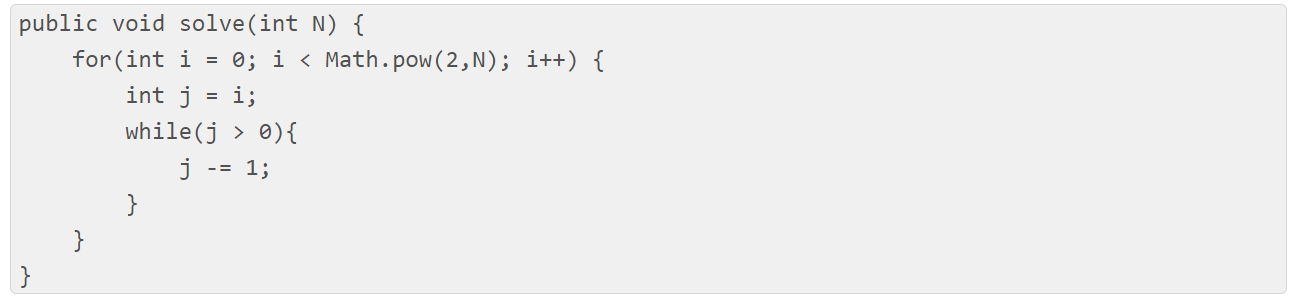
**O(N) time, O(N) space**

**O(N) time, O(1) space**

**O(N \* N) time, O(N) space**

**O(N \* N \* N) time, O(1) space**

What is the time complexity of the following code :



**O(N \* N)**

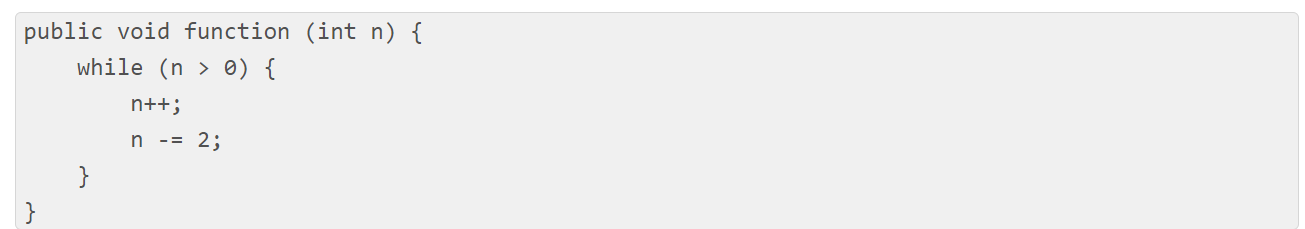
**O(2^N)**

**O(N \* (2^N))**

**O(3^N)**

**O(4^N)**

What will be the time complexity of the above function where n is a positive integer?



**Infinite loop**

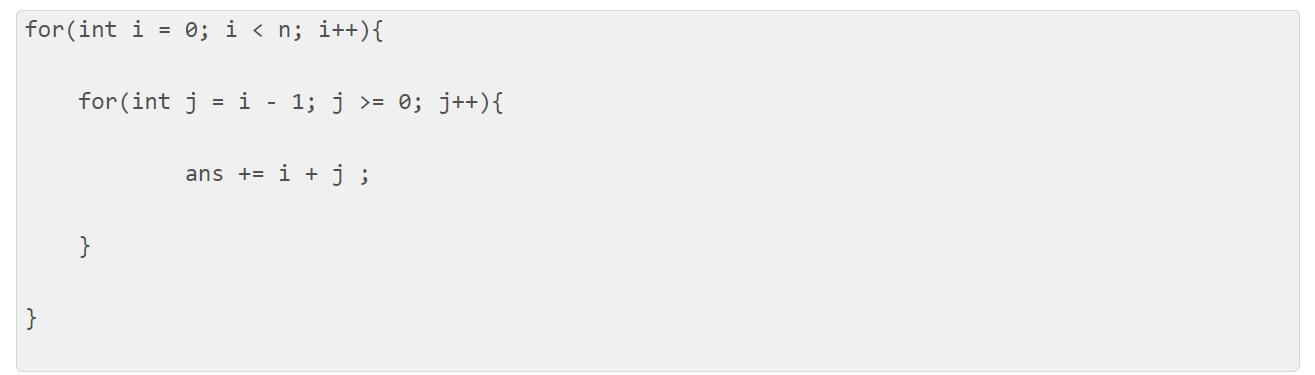
**O(n)**

**O(nlogn)**

**O(logn)**

**None of the above**

What is the time complexity of the following code :



**O(n)**

**O(n^2)**

**Code will run indefinitely**

**O(1)**

**None of the above**

What is the time complexity of the following code :



**O(N^2)**

**O(N)**

**O(NlogN)**

**None of these**