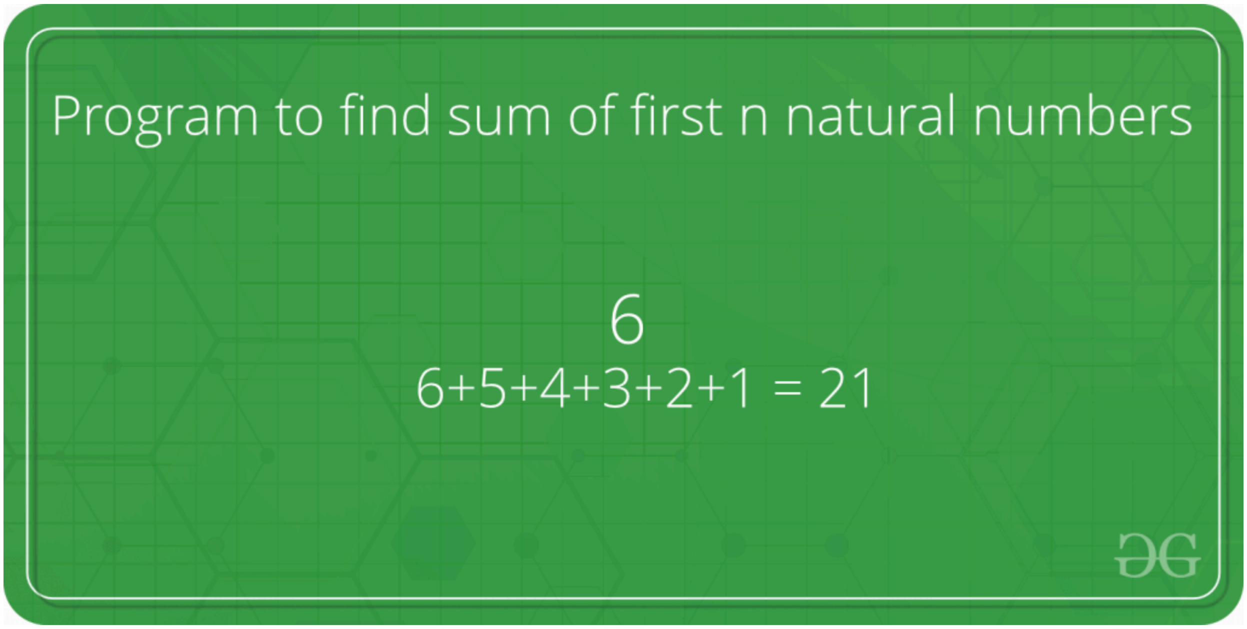
Sum of N Natural Numbers in Java

Given a number n, find the sum of first n natural numbers.



Examples :

Input : n = 3

Output : 6

Explanation :

Note that 1 + 2 + 3 = 6

Input : 5

Output : 15

Explanation :

Note that 1 + 2 + 3 + 4 + 5 = 15

Naive Approach– O(n) Time and O(1) Space

1) Initialize : sum = 0

2) Run a loop from x = 1 to n and

```
  do following in loop.
    sum = sum + x
```

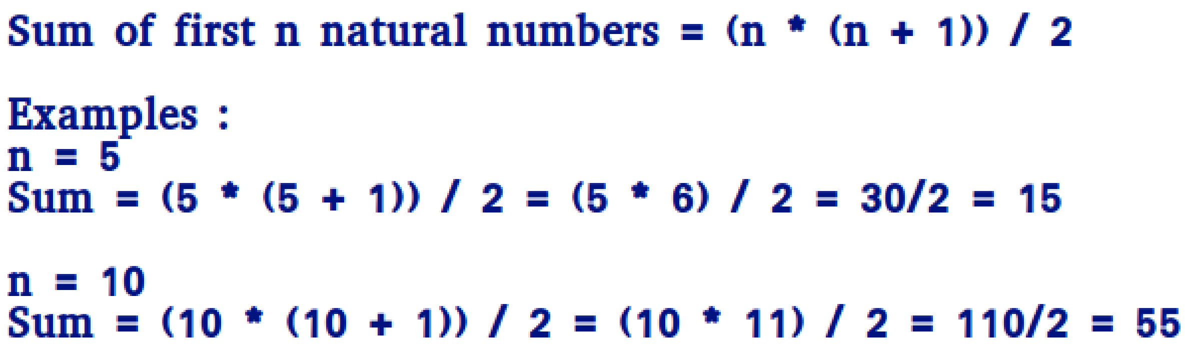
```java
// JAVA program to find sum of first
// n natural numbers.
import java.io.*;

class GfG {
    // Driver code
    public static void main(String args[]) {
        int n = 5;
        int sum = 0;
        for (int x = 1; x <= n; x++)
            sum = sum + x;
        System.out.println(sum);
    }
}
```

```
Output
15
```

Expected Approach – O(1) Time and O(1) Space

An efficient solution is to use the below formula.



Sum of first n natural numbers = (n * (n + 1)) / 2

Examples :
n = 5
Sum = (5 * (5 + 1)) / 2 = (5 * 6) / 2 = 30/2 = 15

n = 10
Sum = (10 * (10 + 1)) / 2 = (10 * 11) / 2 = 110/2 = 55

How does this work?

We can prove this formula using induction.

```
It is true for n = 1 and n = 2
For n = 1, sum = 1 * (1 + 1)/2 = 1
For n = 2, sum = 2 * (2 + 1)/2 = 3

Let it be true for k = n-1.

Sum of k numbers = (k * (k+1))/2
Putting k = n-1, we get
Sum of k numbers = ((n-1) * (n-1+1))/2
= (n - 1) * n / 2

If we add n, we get,
Sum of n numbers = n + (n - 1) * n / 2
= (2n + n2 - n)/2
= n * (n + 1)/2
```

```java
// Efficient JAVA program to find sum
// of first n natural numbers.
import java.io.*;

class GfG {
    // Driver code
    public static void main(String args[]) {
        int n = 5;
        int sum = n * (n + 1) / 2;
        System.out.println(sum);
    }
}
```

```
Output
15
```

Efficient Approach – O(1) Time and O(1) Space

The above program causes overflow, even if the result is not beyond the integer limit. We can avoid overflow up to some extent by dividing first.

```java
// Efficient JAVA program to find sum of first
// n natural numbers that avoids overflow if
// result is going to be within limits.
import java.io.*;

class GfG {
    public static void main(String args[]) {
        int n = 5;
        int sum;
        if (n % 2 == 0)
            sum = (n / 2) * (n + 1);
        // If n is odd, (n+1) must be even
        else
            sum = ((n + 1) / 2) * n;
        System.out.println(sum);
    }
}
```

Output

```
15
```