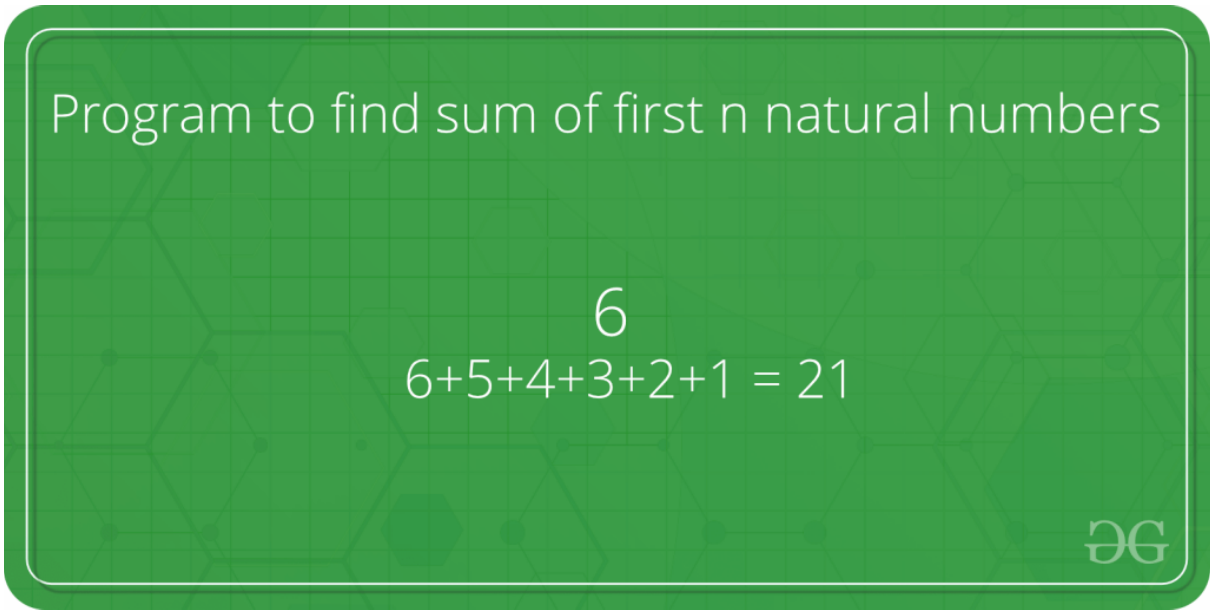
Sum of Natural numbers

Given a number n, find the sum of first 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
```

A **simple solution** is to do the following.

```
1) Initialize : sum = 0
2) Run a loop from x = 1 to n and
   do following in loop.
     sum = sum + x
```

```
// CPP program to find sum of first
// n natural numbers.
#include <iostream>
using namespace std;

// Returns sum of first n natural
// numbersint
findSum(int n)
{
    int sum = 0;
    for (int x = 1; x <= n; x++)
        sum = sum + x;
    return sum;
}

// Driver code
int main()
{
    int n = 5;
    cout << findSum(n);
    return 0;
}
```

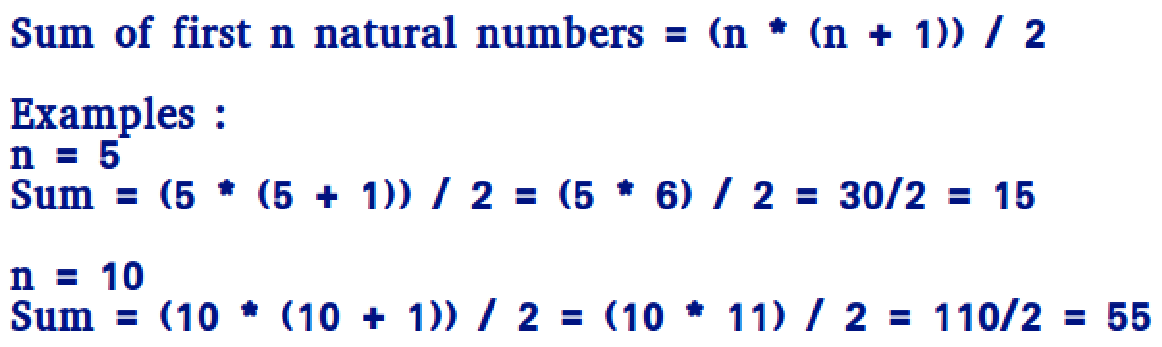
Output

```
15
```

Time Complexity: *O(n)*

Auxiliary Space: *O(1)*

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
```

```
// Efficient CPP program to find sum of first
// n natural numbers.
#include<iostream>
using namespace std;

// Returns sum of first n natural
// numbersint
findSum(int n)
```

```
{
        return n * (n + 1) / 2;
}

// Driver code
int main()
{
        int n = 5;
        cout << findSum(n);
        return 0;
}
```

Output

```
15
```

Time Complexity: *O(1)*

Auxiliary Space: *O(1)*

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.

```
// Efficient CPP program to find sum of first
// n natural numbers that avoids overflow if
// result is going to be within limits.
#include<iostream>
using namespace std;

// Returns sum of first n natural
// numbersint
findSum(int n)
{
        if (n % 2 == 0)

            // Here multiplying by 1LL help to
            // perform calculations in long long,
            // so that answer should not be overflowed
            return (n / 2) * 1LL * (n + 1);

            // If n is odd, (n+1) must be even
            else
            // Here multiplying by 1LL help to
            // perform calculations in long long,
            // so that answer should not be overflowed
            return ((n + 1) / 2) * 1LL * n;
}

// Driver codeint main()
{
        int n = 5;
        cout << findSum(n);
        return 0;
}
```

Output

```
15
```

Time Complexity: *O(1)*

Auxiliary Space: *O(1)*