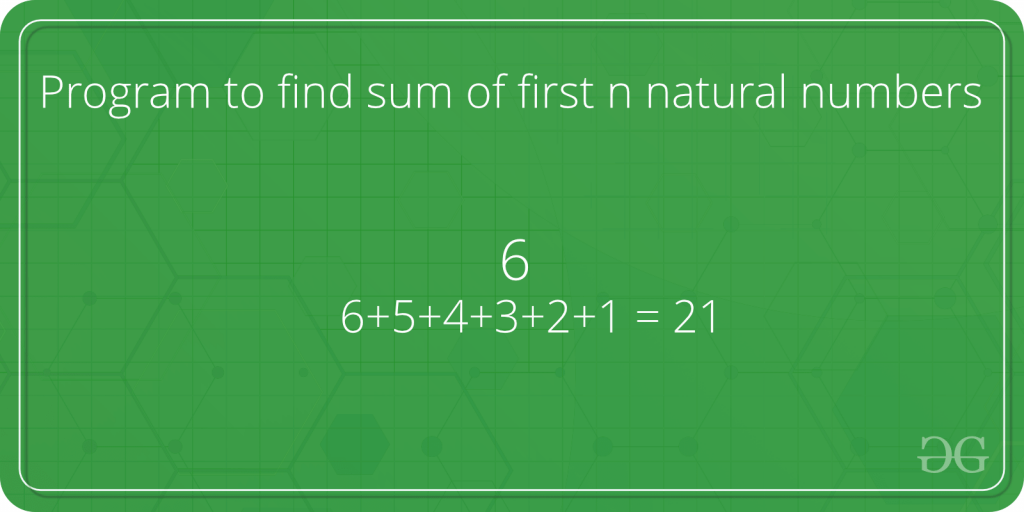
Sum of Natural numbers

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

Python

1

# PYTHON program to find sum of first

2

# n natural numbers.

3

​

4

# Returns sum of first n natural

5

# numbers

6

def findSum(n) :

7

sum = 0

8

x = 1

9

while x <=n :

10

sum = sum + x

11

x = x + 1

12

return sum

13

​

14

​

15

# Driver code

16

​

17

n = 5

18

print (findSum(n))

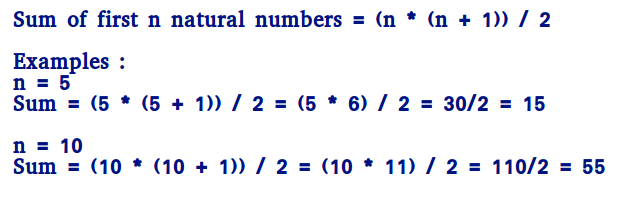
**Output**

15

***Time Complexity:****O(n)*

***Auxiliary Space:****O(1)*

An **efficient solution** is to use the below formula.



**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**

Python

1

# Efficient CPP program to find sum

2

# of first n natural numbers.

3

​

4

# Returns sum of first n natural

5

# numbers

6

def findSum(n) :

7

return n \* (n + 1) / 2

8

9

# Driver code

10

n = 5

11

print (findSum(n))

**Output**

15.0

***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.

Python

# Efficient Python program to find the sum

# of first n natural numbers that avoid

# overflow if the result is going to be

# within limits.

# Returns sum of first n natural

# numbers

def findSum(n) :

if (n % 2 == 0) :

return (n / 2) \* (n + 1)

# If n is odd, (n+1) must be even

else :

return ((n + 1) / 2) \* n

# Driver code

n = 5

print (findSum(n))

**Output**

15.0

***Time Complexity:****O(1)*

***Auxiliary Space:****O(1)*