

Calculate the Nth term

Objective

This challenge will help you learn the concept of recursion.

A function that calls itself is known as a recursive function. The C programming language supports recursion. But while using recursion, one needs to be careful to define an exit condition from the function, otherwise it will go into an infinite loop.

To prevent infinite recursion, *if...else* statement (or similar approach) can be used where one branch makes the recursive call and other doesn't.

```
void recurse() {  
    .....  
    recurse() //recursive call  
    .....  
}  
int main() {  
    .....  
    recurse(); //function call  
    .....  
}
```

Task

There is a series, S , where the next term is the sum of pervious three terms. Given the first three terms of the series, a , b , and c respectively, you have to output the n^{th} term of the series using recursion.

Recursive method for calculating n^{th} term is given below.

$$S(n) = \begin{cases} a & n = 1, \\ b & n = 2, \\ c & n = 3, \\ S(n-1) + S(n-2) + S(n-3) & otherwise \end{cases}$$

Input Format

- The first line contains a single integer, n .
- The next line contains 3 space-separated integers, a , b , and c .

Constraints

- $1 \leq n \leq 20$
- $1 \leq a, b, c \leq 100$

Output Format

Print the n^{th} term of the series, $S(n)$.

Sample Input 0

```
5
1 2 3
```

Sample Output 0

```
11
```

Explanation 0

Consider the following steps:

1. $S(1) = 1$
2. $S(2) = 2$
3. $S(3) = 3$
4. $S(4) = S(3) + S(2) + S(1)$
5. $S(5) = S(4) + S(3) + S(2)$

From steps **1**, **2**, **3**, and **4**, we can say $S(4) = 3 + 2 + 1 = 6$; then using the values from step **2**, **3**, **4**, and **5**, we get $S(5) = 6 + 3 + 2 = 11$. Thus, we print **11** as our answer.