

计算函数值: <https://www.lanqiao.cn/problems/5194/learning> 1星

递归求组合数: <https://www.lanqiao.cn/problems/19880/learning/> 1星

什么是递归

递归 (Recursion) 是一种在函数内部调用自身的方法, 常用于解决具有重复结构的问题。递归的核心思想是将问题拆分为更小的子问题, 直到子问题可以直接求解, 然后再逐步合并结果。

递归的基本要素

1. **终止条件 (Base Case)** : 递归必须有一个明确的终止条件, 否则会无限递归导致栈溢出 (Stack Overflow) 。
2. **递归关系 (Recursive Case)** : 定义问题如何分解成更小的子问题, 并递归调用自身。
3. **递归回溯 (Return Process)** : 当到达终止条件后, 递归逐层返回并合并结果。

递归求阶乘

```
1 int f(int n){
2     if(x==1) return 1;
3     return f(n-1)*n;
4 }
```

递归求斐波拉契序列

```
1 int f(int n){
2     if(n==1||n==2) return n;
3     return f(n-1)+f(n-2);
4 }
```

计算函数值

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int f(int x){
4     if(x==0) return 1;
5     if(x%2==0) return f(x/2);
6     return f(x-1)+1;
7 }
8 int main(){
9     int n;
10    scanf("%d",&n);
11    printf("%d",f(n));
12
13 }
```

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static int f(int x) {
5         if (x == 0) return 1;
```

```

6         if (x % 2 == 0) return f(x / 2);
7         return f(x - 1) + 1;
8     }
9
10    public static void main(String[] args) {
11        Scanner scanner = new Scanner(System.in);
12        int n = scanner.nextInt();
13        System.out.println(f(n));
14        scanner.close();
15    }
16 }
17

```

```

1 def f(x):
2     if x == 0:
3         return 1
4     if x % 2 == 0:
5         return f(x // 2)
6     return f(x - 1) + 1
7
8 n = int(input())
9 print(f(n))

```

递归求组合数

组合数的递推式为：

$$C_n^m = C_{n-1}^{m-1} + C_{n-1}^m$$

递归边界为 $C_{n,0} = C_{n,n} = 1$

```

1 int C(int n,int m){
2     if(n==m||m==0) return 1;
3     return ???;
4 }

```