

组合数 C_n^m 表示从 n 个元素中选取 m 个元素的方案数，其递推式和性质如下：

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

1. 组合数的定义

组合数 C_n^m 的公式为：

$$C_n^m = \frac{n!}{m!(n-m)!}$$

其中：

- $n!$ 表示 n 的阶乘。
- $m!$ 表示 m 的阶乘。
- $(n-m)!$ 表示 $(n-m)$ 的阶乘。

2. 递推公式

组合数满足以下递推关系：

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

递推公式的直观解释：

- **选第 n 个元素**：从剩下的 $n-1$ 个元素中选 $m-1$ 个元素，方案数为 C_{n-1}^{m-1} 。
- **不选第 n 个元素**：从剩下的 $n-1$ 个元素中选 m 个元素，方案数为 C_{n-1}^m 。
- 总方案数为两者之和。

3. 边界条件

- 当 $m=0$ 或 $m=n$ 时， $C_n^m = 1$ 。

```
1  #include <iostream>
2  using namespace std;
3  int C(int n,int m){
4      if(n==m||m==0) return 1;
5      return C(n-1,m-1)+C(n-1,m);
6  }
7  int main()
8  {
9      int t;
10     cin>>t;
11     while(t-->0){
12         int n,m;
13         cin>>n>>m;
14         cout<<C(n,m)<<"\n";
15     }
16     return 0;
17 }
```

```
1  import java.util.Scanner;
2
3  public class Main {
4      // 计算组合数 C(n, m)
```

```

5     public static int C(int n, int m) {
6         if (n == m || m == 0) {
7             return 1; // 边界条件
8         }
9         return C(n - 1, m - 1) + C(n - 1, m); // 递推公式
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14        int t = sc.nextInt(); // 输入测试用例数量
15        while (t-- > 0) {
16            int n = sc.nextInt(); // 输入 n
17            int m = sc.nextInt(); // 输入 m
18            System.out.println(C(n, m)); // 输出 C(n, m)
19        }
20    }
21 }

```

```

1  # 计算组合数 C(n, m)
2  def C(n, m):
3      if n == m or m == 0:
4          return 1 # 边界条件
5      return C(n - 1, m - 1) + C(n - 1, m) # 递推公式
6
7  def main():
8      t = int(input()) # 输入测试用例数量
9      for _ in range(t):
10         n, m = map(int, input().split()) # 输入 n 和 m
11         print(C(n, m)) # 输出 C(n, m)
12
13  if __name__ == "__main__":
14      main()

```