

资源限制

内存限制：256.0MB

C/C++ 时间限制：1.0s

Java 时间限制：3.0s

Python 时间限制：5.0s

问题描述

在一个 $n * n$ 的棋盘上，每个格子中至多放置一个车，且要保证任何两个车都不能相互攻击，有多少种放法（车与车之间是没有差别的）

输入格式

包含一个正整数 n

输出格式

一个整数，表示放置车的方法数

样例输入

2

样例输出

7

数据规模与约定

$n \leq 8$

【样例解释】一个车都不放为1种，放置一个车有4种，放置2个车有2种。

解题思路

这道题是八皇后的阉割版，我们只需要用DFS暴力搜索出所有可能的放置情况即可。

代码如下：

```
import java.util.*;
import java.io.*;

public class Main{
    static int N = 10;
    static int n;
    static boolean[] row = new boolean[N];
```

```

static boolean[] col = new boolean[N];
static int ans = 0;

static void dfs(int x, int y, int sum) {
    if (y == n) {
        y = 0;
        x += 1;
    }
    if (x == n) {
        if (sum == 0) {
            ans ++ ;
        }
        return;
    }
    dfs(x, y+1, sum);

    if (!row[x] && !col[y]) {
        row[x] = col[y] = true;
        dfs(x, y+1, sum-1);
        row[x] = col[y] = false;
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    PrintWriter pw = new PrintWriter(System.out);
    n = sc.nextInt();
    int res = 0;
    for (int i = 0; i <= n; i ++ ) {
        ans = 0;
        Arrays.fill(row, false);
        Arrays.fill(col, false);
        dfs(0, 0, i);
        res += ans;
    }
    pw.println(res);
    pw.close();
}
}

```