DP #2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 种植 | 计数 | 棋盘 | 树 |
| 输入文件名 | plant.in | count.in | chess.in | tree.in |
| 输出文件名 | plant.out | count.out | chess.out | tree.out |
| 每个测试点时限 | 1s | 3s | 2s | 1 s |
| 测试点数目 | 10 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 | 10 |
| 内存限制 | 256M | 256M | 256M | 256M |
| 是否有部分分 | 否 | 否 | 否 | 否 |
| 题目类型 | 传统 | 传统 | 传统 | 传统 |

注意：代码长度限制均为64K，不开O2。

1 **种植**(plant.c/cpp/pas)

1.1 **题目描述**

现在有一块 n 行 m 列的地，你想要在上面种植你心爱的花，为了避免它们争夺肥料，你希望任意两朵花不能上下左右紧邻。另外一些地方杂草丛生，所以你不能在这些位置种上你的花。

现在你希望知道有多少种种花的方案，注意什么都不种也是一种方案。

1.2 **输入格式**

第一行为两个整数n ，m，表示行和列。

接下来 n 行，每行 m 个数，若为0表示该位置不能种花，为 1 表示能种花。

1.3 **输出格式**

输出一行一个整数，表示方案数对 100000000 取模的结果。

1.4 **样例输入**

2 3

1 1 1

0 1 0

1.5 **样例输出**

9

1.6 **数据范围与约定**

对于20％的数据，n , m<= 5。

对于40％的数据，n , m<= 10。

对于100％的数据，n, m<=12。

2 **计数** (count.c/cpp/pas)

2.1 **题目描述**

给出 L,R，求 [L,R] 之中各位数字之和能整除原数的数个数。

2.2 **输入格式**

一行两个整数，分别表示L, R。

2.3 **输出格式**

输出一行一个数，表示答案。

2.4 **样例输入**

10 19

2.5 **样例输出**

3

2.6 **数据范围与约定**

对于20％的数据，保证 R<=100000。

对于100％的数据，保证1 <= L <= R <= 1018。

3 **棋盘**(chess.c/cpp/pas)

3.1 **题目描述**

你现在要在棋盘上放棋子，棋盘规格为 n\*m，需要满足任意一个 n\*n 的区域内都有 C 个棋子。请输出有多少个满足条件的方案。

3.2 **输入格式**

一行三个整数n, m, C，含义如题所述。

3.3 **输出格式**

输出一行一个整数，表示答案对 109+7 取模的结果。

3.4 **样例输入**

2 3 1

3.5 **样例输出**

6

3 .6 **数据范围与约定**

对于20％的数据，n, K<=4；

对于另外20％的数据，m=n；

对于另外20％的数据，n<=50；

对于100％的数据，1<=n<=100；1<=m<=1018；1<=C<=n2

4 **树**(tree.c/cpp/pas)

4.1 **题目描述**

现在有一棵 n 个点的树，每个点颜色非黑即白，有Q次询问，每次给出x, y，询问是否存在一个 x 个点的联通子图，其中黑点数目为 y。

4.2 **输入格式**

第一行两个整数， n 和 Q ，分别表示树的节点数和询问次数。

接下来 n-1 行，每行两个数a，b，表示a、b之间有一条边。

接下来一行有 n 个用空格隔开的整数，第i个数若为1，则表示第 i 个点为白色，否则为黑色。

接下来 Q 行，每行两个用空格隔开的整数 x 和 y，表示询问 。

4.3 **输出格式**

输出Q行，每行为“YES”表示存在，或“NO”表示不存在。

4.4 **样例输入**

9 4  
4 1  
1 5  
1 2  
3 2  
3 6  
6 7  
6 8  
9 6  
0 1 0 1 0 0 1 0 1  
3 2  
7 3  
4 0  
9 5

4.5 **样例输出**

YES

YES

NO

NO

4.6 **数据范围与约定**

对于20％的数据，保证 n <= 10。

对于50％的数据，保证1<=n<=500。

对于100％的数据，保证 1<=n<=5000，q<=10^5