

# NOIp Senior Day1

by Timber

题目名称	color	haiku	flower
目录	color	haiku	flower
可执行文件名	color	haiku	flower
输入文件名	color.in	haiku.in	flower.in
输出文件名	color.out	haiku.out	flower.out
时间限制	1s	1s	1s
空间限制	512MB	512MB	512MB
测试点数量	10	20	20
测试点分数	10	5	5
比较方式	全文	全文	全文
部分分	无	无	无

提交源程序文件名

对于C++ 语言	color.cpp	haiku.cpp	flower.cpp
对于C 语言	color.c	haiku.c	flower.c
对于Pascal 语言	color.pas	haiku.pas	flower.pas

编译选项

对于C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于C 语言	-O2	-O2	-O2
对于Pascal 语言	-O2	-O2	-O2

## 1 color

### 1.1 题目描述

给定一个 $H$ 行 $W$ 列的矩阵，矩阵中的点一开始均为白色，然后将指定的 $N$ 个点染成黑色，询问有多少个 $3 \times 3$ 的区域中有恰好 $i$ 个黑色点 $0 \leq i \leq 9$ 。

### 1.2 输入格式(color.in)

输入共 $N + 1$ 行，第一行三个整数，分别是 $H, W, N$ ，含义如题目所示。接下来 $N$ 行，每行两个整数 $x, y$ ，表示将坐标为 $x, y$ 的点染成黑色。

### 1.3 输出格式(color.out)

输出共10行，每行一个数，第 $i$ 行代表有多少个 $3 \times 3$ 的区域中恰好有 $i - 1$ 个黑色点。

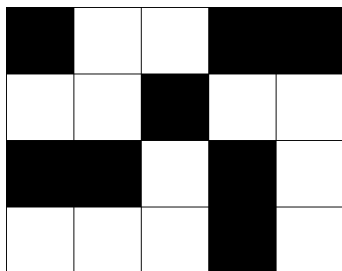
### 1.4 输入样例

详见sample/color/color.in。

### 1.5 输出样例

详见sample/color/color.out。

### 1.6 样例解释



### 1.7 数据范围

$$3 \leq H, W \leq 10^9。$$

$$0 \leq N \leq \min(10^5, H \times W)。$$

$$1 \leq x_i, y_i \leq N。$$

保证没有任何一个点被重复染色。

## 1.8 部分分

对于40%的数据,  $N \leq 500$ 。

对于100%的数据,  $N \leq 10^5$ 。

其中数据保证有40%的数据  $H, W \leq 10^5$ 。

## 2 haiku

### 2.1 题目描述

给定一个长度为 $N$ 的序列 $a$ ，序列中的每个数都可以是 $1 \sim 10$ 中的某一个，总共有 $10^N$ 种序列，现在给出 $N, X, Y, Z$ ，问有多少个长度为 $N$ 的序列满足其有四个下标 $x, y, z, w$ 使得 $a[x] + a[x + 1] + \dots + a[y - 1] = X$ ， $a[y] + a[y + 1] + \dots + a[z - 1] = Y$ ， $a[z] + a[z + 1] + \dots + a[w] = Z$ 。

### 2.2 输入格式(haiku.in)

输入共一行，四个整数，分别为 $N, X, Y, Z$ ，含义如题目所示。

### 2.3 输出格式(haiku.out)

输出共一行，一个整数，表示答案对 $10^9 + 7$ 的值。

### 2.4 输入样例

详见sample/haiku/haiku.in。

### 2.5 输出样例

详见sample/haiku/haiku.out。

### 2.6 样例解释

在这组样例中，只存在一组合法的序列，即 $a[] = [5, 7, 5]$ 。

### 2.7 数据范围

$$3 \leq N \leq 40。$$

$$1 \leq X \leq 5, 1 \leq Y \leq 7, 1 \leq Z \leq 5。$$

### 2.8 部分分

对于30%的数据， $N \leq 5$ 。

对于50%的数据， $N \leq 8$ 。

对于80%的数据， $N \leq 20$ 。

对于100%的数据， $N \leq 40$ 。

## 3 flower

### 3.1 题目描述

艺术馆门前将摆出许多花，一共有  $n$  个位置排成一排，每个位置可以摆花也可以不摆花。有些花如果摆在相邻的位置（隔着一个空的位置不算相邻），就不好看了。假定每种花数量无限，求摆花的方案数。

### 3.2 输入格式(flower.in)

输入有  $1 + m$  行，第一行有两个用空格隔开的正整数  $n$ 、 $m$ ， $m$  表示花的种类数。接下来的  $m$  行，每行有  $m$  个字符 1 或 0，若第  $i$  行第  $j$  列为 1，则表示第  $i$  种花和第  $j$  种花不能排在相邻的位置，输入保证对称。（提示：同一种花可能不能排在相邻位置）。

### 3.3 输出格式(flower.out)

输出只有一个整数，为方案数（这个数字可能很大，请输出方案数除以  $10^9 + 7$  的余数）。

### 3.4 输入样例

详见sample/flower/flower.in。

### 3.5 输出样例

详见sample/flower/flower.out。

### 3.6 样例解释

七种方案为（空，空）、（空，1）、（1，空）、（2，空）、（空，2）、（1，1）、（2，2）。

### 3.7 数据范围

对于所有数据， $n \leq 10^7$ ， $m \leq 100$

### 3.8 部分分

对于30%的数据， $n \leq 5$ ， $m \leq 10$ 。

对于60%的数据， $n \leq 200$ ， $m \leq 100$ 。

对于100%的数据， $n \leq 10^7$ ， $m \leq 100$ 。