

## 2018-7-16 NOIP 提高组模拟赛

题目名	骑士的移动	数星星	$2^k$ 进制数
源文件	a.cpp/c/pas	stars.cpp/c/pas	digital.cpp/c/pas
输入文件	a.in	stars.in	digital.in
输出文件	a.out	stars.out	digital.out
时间限制	1000MS	1000MS	1000MS
内存限制	256MB	256MB	256MB
测试点	5	10	10
测试点分值	20	10	10

## Problem 1 骑士的移动

(a. cpp/c/pas)

### 【题目描述】

Somurolov 先生，确实是一位神话般的国际象棋游戏玩家，断言除了他之外没有其他人能够如此快速地将骑士从一个位置移动到另一个位置。你能打败他吗？你的任务是编写一个程序来计算骑士从另一个点到达一个点所需的最小移动次数，这样你就有机会比 Somurolov 更快。对于不熟悉国际象棋的人来说，可能的骑士动作如图 1 所示。

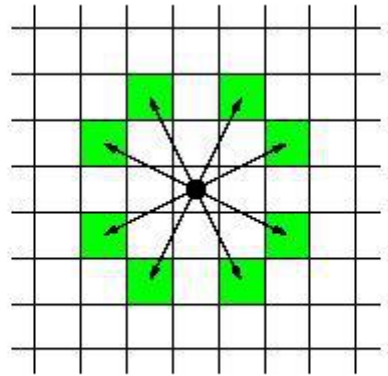


Figure 1: Possible knight moves on the board.

### 【输入数据】

测试样例数量  $n$ ，接下来是  $n$  个样例。每个样例由以下整数组成：  
第一行指定棋盘一侧的长度  $l$  ( $4 \leq l \leq 300$ )。整个板的尺寸为  $l \times l$ 。  
第二行和第三行包含一对整数  $\{0, \dots, l-1\} \times \{0, \dots, l-1\}$ ，用于指定骑士在棋盘上的起始位置和结束位置。整数由一个空格分隔。您可以假设这些位置是该场景的棋盘上的有效位置。

### 【输出数据】

对于输入的两个测试样例，您必须计算从起点移动到终点所需的最小

骑士移动量。如果起点和终点相等，则距离为零。距离必须写在一行上。

**【样例输入 1】**

```
3
8
0 0
7 0
100
0 0
30 50
10
1 1
1 1
```

**【样例输出 1】**

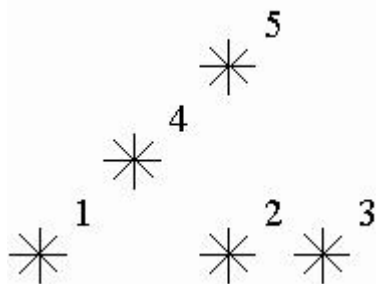
```
5
28
0
```

## Problem2 数星星

(stars.cpp/c/pas)

### 【题目描述】

天文学家经常研究星形图，其中恒星由平面上的点表示，每颗恒星都有笛卡尔坐标。让恒星的水平为不高于恒星的数量，而不是给定恒星的右侧。天文学家想知道恒星水平的分布。



例如，查看上图中显示的地图。星号 5 的等级等于 3（它由三颗恒星形成，数字为 1,2 和 4）。并且 2 和 4 编号的星的等级是 1。在该地图上，0 级的恒星有 1 颗，1 级的恒星有 2 颗，2 级的恒星和 3 级的恒星各 1 颗。您将编写一个程序，用于计算给定地图上每个级别的星星数量。

### 【输入数据】

输入的第一行包含许多分  $N$  ( $1 \leq N \leq 15000$ )。下面  $N$  行描述（两个整数的分坐标  $x$  和  $y$ ,  $0 \leq x, y \leq 32000$ ）。一个点上只能有一颗恒星。星号按  $y$  坐标的升序列出。具有相等  $y$  坐标的星以  $x$  坐标的升序列出。

### 【输出数据】

输出应包含  $n$  行，每行一个数字。第一行包含等级 0 的恒星数量，第二行包含等级 1 的恒星数量，依此类推，最后一行包含等级  $n-1$  的恒星数量。

**【样例输入 1】**

```
5
1 1
5 1
7 1
3 3
5 5
```

**【样例输出 1】**

```
1
2
1
1
0
```

### Problem 3 $2^k$ 进制数

(digital.cpp/c/pas)

#### 【题目描述】

设  $r$  是个  $2^k$  进制数，并满足以下条件：

- (1)  $r$  至少是个 2 位的  $2k$  进制数。
- (2) 作为  $2k$  进制数，除最后一位外， $r$  的每一位严格小于它右边相邻的那一位。
- (3) 将  $r$  转换为 2 进制数  $q$  后，则  $q$  的总位数不超过  $w$ 。

在这里，正整数  $k$  ( $1 \leq k \leq 9$ ) 和  $w$  ( $k < w \leq 30000$ ) 是事先给定的。

问：满足上述条件的不同的  $r$  共有多少个？

我们再从另一角度作些解释：设  $S$  是长度为  $w$  的 01 字符串（即字符串  $S$  由  $w$  个“0”或“1”组成）， $S$  对应于上述条件（3）中的  $q$ 。

将  $S$  从右起划分为若干个长度为  $k$  的段，每段对应一位  $2k$  进制的数，如果  $S$  至少可分成 2 段，则  $S$  所对应的二进制数又可以转换为上述的  $2k$  进制数  $r$ 。

例：设  $k=3$ ， $w=7$ 。则  $r$  是个八进制数（ $2^3=8$ ）。由于  $w=7$ ，长度为 7 的 01 字符串按 3 位一段分，可分为 3 段（即 1，3，3，左边第一段只有一个二进制位），则满足条件的八进制数有：

2 位数：高位为 1：6 个（即 12，13，14，15，16，17），高位为 2：5 个， $\dots$ ，高位为 6：1 个（即 67）。共  $6+5+\dots+1=21$  个。

3 位数：高位只能是 1，第 2 位为 2：5 个（即 123，124，125，126，127），第 2 位为 3：4 个，…，第 2 位为 6：1 个（即 167）。共  $5+4+\cdots+1=15$  个。所以，满足要求的  $r$  共有 36 个。

**【输入数据】**

只有 1 行，为两个正整数，用一个空格隔开：

$k$     $W$

**【输出数据】**

输出为 1 行，是一个正整数，为所求的计算结果，即满足条件的不同的  $r$  的个数（用十进制数表示），要求最高位不得为 0，各数字之间不得插入数字以外的其他字符（例如空格、换行符、逗号等）。

（提示：作为结果的正整数可能很大，但不会超过 200 位）

**【样例输入 1】**

3   7

**【样例输出 1】**

36