“Pony.ai 杯”2018 清华大学
学生程序设计竞赛暨高校邀请赛

时间：2018 年 5 月 13 日 10:00 ~ 15:00

目 录

A. 绿绿与串串 / **String** **2**

B. 赛艇 / **Citing** **4**

C. 好图计数 / **Count** **6**

D. 《算法与数据结构》小助教招募通知 / **Dsa** **8**

E. 淘米神的树 / **Tommy** **10**

F. 密码学第三次小作业 / **Rsa** **11**

G. 琪亚娜之墙 / **Wall** **13**

H. 组合数问题 / **Problem** **15**

I. 城市地铁规划 / **City** **23**

J. 弗雷兹的玩具商店 / **Toyshop** **25**

K. 蛋糕 / **Cake** **28**

L. 生生不息 / **Lives** **30**

M. 为最美的那位献上花朵 / **M** **32**

A. 绿绿与串串 / String

时间限制： 2.0 秒

空间限制： 512 MB

【题目背景】

绿绿和 Yazid 是好朋友。他们在一起做串串游戏。

【题目描述】

绿绿有一个由小写字母组成的非空字符串 R，但 Yazid 不知道它具体是什么。

我们定义**翻转**的操作：把一个串以最后一个字符作对称轴进行翻转复制。形式化地描述就是，如果他翻转的串为 R，那么他会将前 $|R|-1$ 个字符倒序排列后，插入到串的最后。

举例而言，串abcd进行翻转操作后，将得到abcdcba；串qw连续进行 2 次翻转操作后，将得到qwqwq；串z无论进行多少次翻转操作，都不会被改变。

贪玩的绿绿进行了若干次（可能为 0 次）翻转操作。

淘气的绿绿又展示出了一个非空串 S，并表示 S 是**最终**的串 R 的前缀。现在，他想考考 Yazid，**初始**的串 R 的长度可能是多少。

Yazid 找到了正在参加清华校赛的你，请你来帮他解决这个问题。但聪明的 Yazid 发现，所有超过 $|S|$ 的整数都一定是 R 的可能长度，因此你只需要告诉他不超过的 $|S|$ 的 R 的可能长度即可。

为了帮助你理解问题，Yazid 还将对一些概念和记号做出解释：

- 对于一个串 S，$|S|$ 表示的是该串的长度。
- 对于一个串 S，我们定义串 T 是它的前缀，当且仅当 $|T| \le |S|$，且对于任意整数 i 满足 $1 \le i \le |T|$，都有 T 的左起第 i 个字符与 S 的左起第 i 个字符相同。（形象地理解，即 T 在 S 的前部出现）
  - 如：abc是abcdefg的前缀，aba不为abba的前缀，z为z的前缀，空串为任意一个串的前缀。

【输入格式】

从标准输入读入数据。

输入包含多组数据，第一行一个整数 T 表示数据组数。接下来依次描述每组数据，对于每组数据：

- 一行一个仅由小写字母组成的非空字符串 S。

【输出格式】

输出到标准输出。

对于每组数据，输出 1 行：

- 从小到大输出 $|R|$ 的所有不超过 $|S|$ 的可能值，所有值之间用单个空格隔开。

【样例 1 输入】

```
4
abcdcb
qwqwq
qaqaqqq
carnation
```

【样例 1 输出】

```
4 6
2 3 4 5
6 7
9
```

【数据规模】

保证 $|S| \le 10^6$，$\sum |S| \le 5 \times 10^6$。

$\sum |S|$ 表示的是单个测试点中所有数据 $|S|$ 的总和。

【提示】

- 读入规模较大，请注意读入效率。
- 样例中的最后一个字符串是什么意思呢？

B. 赛艇 / Citing

时间限制： 12.0 秒

空间限制： 2048 MB

【题目描述】

Lavender、Caryophyllus、Jasmine、Dianthus 现在在玩一款名叫 “赛艇” 的游戏。这个游戏的规则是这样的：

1. 玩家自由组成两队，一个人当赛艇的艇长，另一个人当侦察兵；
2. 每次游戏开始时，双方均拥有由系统生成的某张地图，该地图以 01 矩阵的形式表示，1表示有障碍物，无法通行，0表示水域空旷，可以通行；
3. 第一回合，双方的赛艇艇长都要在地图上指定一个出发点，该出发点不能是障碍物，也就是只能为0；
4. 在每个回合中，艇长可以指挥自己的赛艇向上/下/左/右四个方向的某一方向的空旷水域移动一个单位的距离，也就是说只能移向四个方向上的某个0上（当然，不能移动出地图之外）；在该操作完成之后，**必须向对方说出自己在该回合移动的方向**；
5. 双方的侦察兵负责记录每一回合对方赛艇的移动方向，并负责推断此时对方赛艇可能的位置；如果某方的侦察兵推测出对方赛艇此时的精确位置，那么可以向其发射导弹，该侦察兵所在的一方胜利；

现在，Jasmine 记录了一些对方赛艇的路径，她想确定一下此时对方所有可能的位置共有几种。由于她不是很擅长计算，所以这个任务就交给你了。

【输入格式】

从标准输入读入数据。

输入第一行包含三个正整数 n，m，k，分别表示地图为 n 行 m 列，当前游戏已经进行了 k 轮。保证 $2 \le n, m \le 1500$，$1 \le k \le 5 \times 10^6$。

输入第二行到第 $n+1$ 行为一个 n 行 m 列的 01 矩阵，无任何分隔符号，表示地图的具体信息，具体含义如上所示。

输入的最后一行为一个长度为 k 的字符串 s，仅由字母 w、a、s、d 构成，从前往后第 i 个字符 s_i 表示对方在第 i 轮中，对方赛艇向上/左/下/右移动一个单位距离。

【输出格式】

输出到标准输出。

输出一行一个正整数，表示在第 k 轮游戏回合的时候，对方赛艇可能的位置的种数。对于所有输入数据，**保证有合法解**。

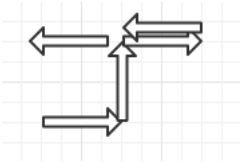
【样例 1 输入】

```
5 6 5
000000
001001
000100
001000
000001
dwdaa
```

【样例 1 输出】

```
4
```

【样例 1 解释】



上图显示了路径序列可视化之后的结果，下图用蓝色标出了此时对方赛艇可能的位置。

000000
001001
000100
001000
000001

C. 好图计数 / Count

时间限制： 1.0 秒

空间限制： 512 MB

【题目背景】

这道题目非常简单，它甚至没有题目背景、没有任何故事。

但为了能让你顺利理解题目，善良的 Yazid 将为你介绍一些概念。

- 简单图：不存在重边、自环的图。（重边即为两条完全相同的边，自环即为两端点为同一节点的边）
- 补图：一个图 G 的补图有与 G 完全相同的节点，且任意两点之间有边当且仅当他们在 G 中不相邻。

【题目描述】

我们归纳定义一个无向简单图是**好的**：

1. 一个单点是好的。
2. 若干个好的图分别作为联通块所形成的图是好的。
3. 一个好的图的补图是好的。

给定一个正整数 n。

求 n 个点的本质不同的好的图的数量对质数 P 取模的结果。（这里的 P 是一个常数，具体见【输入格式】）

两个好的图的被认为是**本质不同的**，当且仅当无论如何将一个图重标号，它都不能与另一个图完全相同。

【输入格式】

从标准输入读入数据。

输入包含多组数据，第一行 2 个用空格隔开的整数 T, P 分别表示数据组数、以及模数。接下来依次描述每组数据，对于每组数据：

- 一行一个整数 n，表示希望统计数目的好的图的节点数。

【输出格式】

输出到标准输出。

对于每组数据，输出 1 行：

- 一个整数，表示 n 个节点的好的图的数目对 P 取模的结果。

【样例 1 输入】

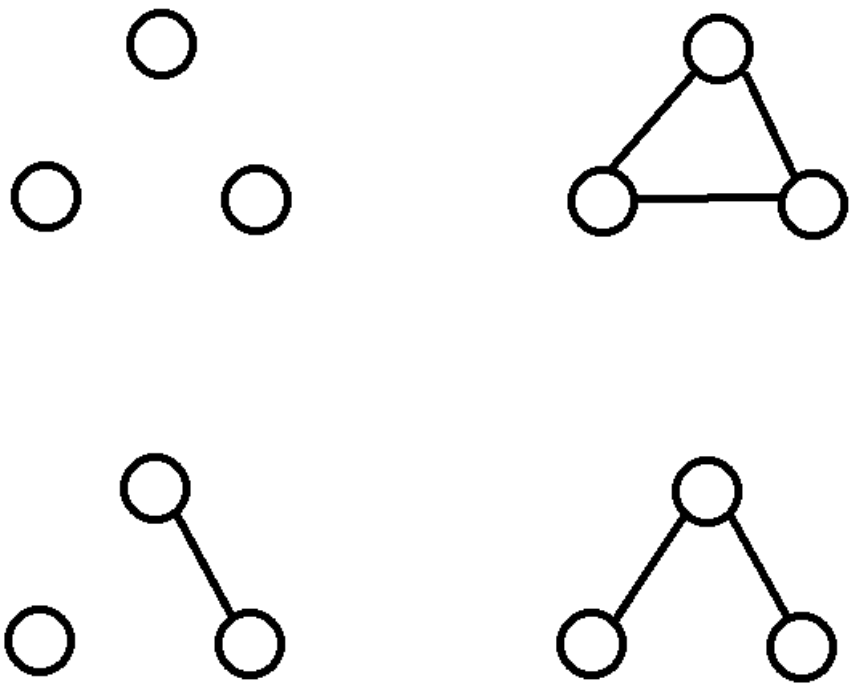
```
2 233
3
4
```

【样例 1 输出】

```
4
10
```

【样例 1 解释】

下面是 3 个点的所有好的图：



【数据规模】

保证 $T \le 233$，$n \le 23333$，$2^{29} < P < 2^{30}$ 且保证 P 为质数。

【提示】

能够通过本题的算法的时间复杂度可能比你想象的要糟糕一些、也可能比你想象的要优秀一些。

D. 《算法与数据结构》小助教招募通知 / Dsa

时间限制： 1.5 秒

空间限制： 512 MB

【题目背景】

T 大学和 U 大学是 W 国最好的两个大学。《算法与数据结构》是 U 大学的一门经典课程，曾连续百余年获得 “U 大学一级精品课程” 的荣誉。

小 O 是这门课明年的助教。由于这门课过于火爆，他一个人忙不过来，所以他正在招募这门课的小助教。不过，招募的条件有点奇怪.....

【题目描述】

《算法与数据结构》课程的内容共分为 n 个知识点，而每个知识点均有配套的的 m 道练习题。每道题有两个属性值，第 i 个知识点的第 j 道练习题可以让做这道题的学生的算法能力提高 $a_{i,j}$，数据结构能力提高 $d_{i,j}$。这些属性值均为整数，但是可能为负数，甚至可能存在两个属性均为负的练习题。（不要问这种题是怎么被加进题库的...）

明年的《算法与数据结构》课共有 q 个学生注册。通过对每个学生以前学习情况的考察，小 O 决定对每个学生进行 “因材施教”。具体方法是，对于第 k 个学生、第 i 个知识点，小 O 希望这个学生完成这个知识点的恰好 $c_{k,i}$ 道练习题。

其实小 O 和 T 大学关系密切，希望在 U 大学搞点事情。对于每个学生，设为该学生选出的所有练习题对于算法能力与数据结构能力的提升分别为 A 与 D，小 O 的目标是最大化 $A^2 + D^2$，这样无论这个学生在哪个方面能力提升或降低的程度很大，都是小 O 所期望的。

请对于每个学生计算这个目标的最大值。

【输入格式】

从标准输入读入数据。

每个输入文件仅包含一个测试数据。

- 第一行包含三个正整数 n, m, q。
- 接下来 n 行，每行包含 m 个整数；其中第 i 行的第 j 个整数为 $a_{i,j}$。
- 接下来 n 行，每行包含 m 个整数；其中第 i 行的第 j 个整数为 $d_{i,j}$。
- 接下来 q 行，每行包含 n 个整数；其中第 k 行的第 i 个整数为 $c_{k,i}$。

对于输入的每一行，如果有多个数输入，则相邻两个数用恰好一个空格隔开。

【输出格式】

输出到标准输出。

输出 q 行，每行 1 个整数：第 k 行的整数表示第 k 个学生的 $A^2 + D^2$ 的最大值。

【样例 1 输入】

```
2 4 4
1 1 -1 -1
0 10 0 -1
1 -1 1 -1
10 0 -1 0
1 1
2 1
1 2
2 2
```

【样例 1 输出】

```
122
144
242
244
```

【样例 1 解释】

有 2 个知识点，每个知识点有 4 道练习题；有 4 个学生。

以第 2 个学生为例，我们需要为该学生在第 1 个知识点中选择恰好 2 道练习题，在第 2 个知识点中选择恰好 1 道练习题。一种最优的方案是，在第 1 个知识点中选择第 1 道题和第 3 道题，在第 2 个知识点中选择第 1 道题。这些题目对于算法能力的提高为 $A = a_{1,1} + a_{1,3} + a_{2,1} = 1 + (-1) + 0 = 0$，对于数据结构能力的提高为 $D = d_{1,1} + d_{1,3} + d_{2,1} = 1 + 1 + 10 = 12$，此时 $A^2 + D^2 = 144$，并且不存在更优的方案。

【数据规模】

$1 \le n \le 6$，$1 \le m \le 666$，$1 \le q \le 666$，所有的 $a_{i,j}$ 与 $d_{i,j}$ 均在 $[-10^9, 10^9]$ 范围内。对于最终测试数据，所有的 $c_{k,i}$ 均在 $[0, m]$ 范围内独立、均匀随机生成。

【提示】

输出的每个数可能非常大，甚至可能超过 64 位整数的范围。

- C++ 语言可以使用`__int128_t`类型表示带符号 128 位整数，或使用`__uint128_t`类型表示无符号 128 位整数；这两种数据类型无法直接进行输入/输出。
- Java 语言可以使用`BigInteger`类。

E. 淘米神的树 / Tommy

时间限制： 10.0 秒

空间限制： 512 MB

【题目描述】

可爱的 Tommy 有一棵树。这棵树上只有点 a 和 b 是黑色，其它的点都是白色。

每次，Tommy 可以将一个黑色 p 的点染成红色，然后把和 p 相邻的所有白色的点染成黑色。最后，所有的点都会被染成红色。

设第 i 个点是第 t_i 个被染成红色的，那么 t_i 是一个 1 到 n 的排列。Tommy 希望你帮他求出，有多少种不同的 t_i。

【输入格式】

从标准输入读入数据。

每个输入文件仅包含一个测试数据。

- 第一行三个用空格隔开的正整数 n, a, b，表示树的点数和初始的黑色点的编号；
- 接下来 $n-1$ 行，每行两个正整数 x, y 描述树上的一条边 (x, y)。

树上的节点从 1 开始编号。

【输出格式】

输出到标准输出。

输出 1 行：

- 一个整数，表示不同的排列 t_i 的个数除以 $998,244,353$ 的余数。

【样例 1 输入】

```
4 1 2
1 2
2 3
3 4
```

【样例 1 输出】

```
4
```

【数据规模】

保证 $a, b \le n \le 234,567$。

F. 密码学第三次小作业 / Rsa

时间限制： 1.0 秒

空间限制： 512 MB

【题目背景】

1977 年，罗纳德 • 李维斯特（Ron Rivest）、阿迪 • 萨莫尔（Adi Shamir）和伦纳德 • 阿德曼（Leonard Adleman）提出了 RSA 加密算法。RSA 加密算法是一种非对称加密算法，其可靠性由极大整数因数分解的难度决定。换言之，对一极大整数做因数分解愈困难，RSA 算法愈可靠。假如有人找到一种快速因数分解的算法的话，那么用 RSA 加密的信息的可靠性就肯定会极度下降。

RSA 的基本原理如下：

- 公钥与私钥的产生

1. 随机选择两个不同大质数 p 和 q，计算 $N = p \times q$
2. 根据欧拉函数性质，求得 $r = \varphi(N) = \varphi(p)\varphi(q) = (p-1)(q-1)$
3. 选择一个小于 r 的正整数 e，使 e 和 r 互质。并求得 e 关于 r 的乘法逆元 d，有 $ed \equiv 1 \pmod{r}$
4. 将 p 和 q 的记录销毁。此时，(N, e) 是公钥，(N, d) 是私钥。

- 消息加密：首先需要将消息 m 以一个双方约定好的格式转化为一个小于 N，且与 N 互质的整数 n。如果消息太长，可以将消息分为几段，这也就是我们所说的块加密，后对于每一部分利用如下公式加密：

$$n^e \equiv c \pmod{N}$$

- 消息解密：利用密钥 d 进行解密

$$c^d \equiv n \pmod{N}$$

【题目描述】

现在有两个用户由于巧合，拥有了相同的模数 N，但是私钥不同。设两个用户的公钥分别为 e_1 和 e_2，**且两者互质**。明文消息为 m，密文分别为：

$$\begin{aligned} c_1 &= m^{e_1} \bmod N \\ c_2 &= m^{e_2} \bmod N \end{aligned}$$

现在，一个攻击者截获了 c_1 ，c_2 ，e_1 ，e_2 ，N ，请帮助他恢复出明文 m 。

【输入格式】

从标准输入读入数据。

输入包含多组数据，第一行一个整数 T 表示数据组数，保证 $1 \le T \le 10^4$ 。接下来依次描述每组数据，对于每组数据：

- 一行包含五个正整数 c_1，c_2，e_1，e_2，N，保证 $2^8 < N < 2^{63}$，N 有且仅有两个素因子，其余数据严格按照上述 RSA 算法生成。

【输出格式】

输出到标准输出。

对于每组数据，输出 1 行：

- 一个非负整数 m，请选手务必在输出时保证 $0 \le m < N$。答案 m 保证与 N 互质。

【样例 1 输入】

```
1
1502992813 2511821915 653507 57809 2638352023
```

【样例 1 输出】

```
19260817
```

G. 琪亚娜之墙 / Wall

时间限制： 3.0 秒

空间限制： 512 MB

【题目描述】

那一天，人类终于回想起，曾经一度被她们支配的恐惧，还有被囚禁于鸟笼中的那份耻辱。

为了守护最珍视之物，Kiana 建立起了 n 座围墙，每一座围墙都闭合形成了一个**凸多边形**，不同的围墙**互不相交**。出于特别的目的，围墙分为两种：用砖瓦和岩石砌成的**工匠之墙**、用树木和荆棘形成的**自然之墙**。

只依靠墙壁并不使人放心，Kiana 还派遣了两类士兵在墙内巡逻：

追魂猎人：他们从小与大自然为伴，可以自由穿越自然之墙，但无法通过工匠之墙。

天马骑士：他们拥有在天空中飞行的能力，可以自由飞越工匠之墙，但由于某些信仰的缘故，无法通过自然之墙。

只依靠墙壁和士兵任然难以让人放心，但 Kiana 还拥有神之力量，在某段时间内，她发挥了 m 次这股力量，以完成两类事件：

1. 将某座工匠之墙转化为自然之墙，或将某座自然之墙转化为工匠之墙。

2. 将一位追魂猎人或天马骑士直接传送到坐标 (x, y) 处，保证这个坐标不位于围墙之上，且与最近的围墙相距至少 10^{-3}。

在每一次传送士兵完成后，Kiana 希望知道这名士兵可以自由活动的区域**面积**有多大。由于她不会做，所以希望你来告诉她答案。

【输入格式】

从标准输入读入数据。

输入第一行包含两个正整数 n 和 m，分别表示围墙的数量和 Kiana 发动神之力量的次数，数据保证 $n \le 3 \times 10^4$ 且 $m \le 10^5$。

接下来 n 行，第 i 行用于描述第 i 座围墙的情况，首先有两个正整数 k_i 和 r_i，分别表示这座围墙的顶点数和类型，若 $r_i = 1$ 表示这是工匠之墙，若 $r_i = 2$ 表示这是自然之墙，接下来 $2k_i$ 个实数，第 $2p-1$ 和第 $2p$ 个数表示这座围墙上第 p 个顶点的横、纵坐标，所有的顶点按照在多边形上逆时针排列的顺序输入，数据保证 $3 \le k_i \le 100$ 且 $\sum_{i=1}^{n} k_i \le 10^5$。

接下来 m 行，第 j 行用于描述 Kiana 第 j 次使用神之力量的情况，首先有一个正整数 op，表示 Kiana 发动的神之力量类型，若：

$op = 1$：这一行还有一个正整数 u，表示 Kiana 转换了第 u 座围墙的类型（若本来为工匠之墙，则转化为自然之墙，若本来为自然之墙，则转化为工匠之墙），数据保证 $1 \le u \le n$。

$op = 2$：这一行还有一个正整数 v 与两个实数 x 和 y，若 $v = 1$，表示 Kiana 将一名追魂猎人传送到了 (x, y) 处，若 $v = 2$，表示 Kiana 将一名天马骑士传送到了 (x, y) 处。

数据保证所有的 r_i、op 和 v 的取值仅为 1 或 2，所有的实数保留到小数点后两位且绝对值小于 2×10^5，任意两个的多边形之间的距离至少为 10^{-3}，任意两个输入顶点的横坐标之差与纵坐标之差的绝对值不小于 10^{-3}。

【输出格式】

输出到标准输出。

输出由若干行组成，对于每个 $op = 2$ 的输入，输出一行一个实数 S（保留到小数点后 4 位），表示该名士兵可以自由活动的区域面积，数据保证这个面积是有限的。

【样例输入】

```
3 5
4 1 -10.10 -10.20 10.30 -10.40 10.50 10.60 -10.70 10.80
4 2 -20.80 -20.70 20.60 -20.50 20.40 20.30 -20.20 20.10
4 1 -30.10 -30.30 30.50 -30.70 30.20 30.40 -30.60 30.80
2 1 15.80 15.60
2 1 25.40 25.20
1 2
2 1 15.70 15.50
2 1 25.30 25.10
```

【样例输出】

```
3271.8500
3271.8500
1236.0000
2035.8500
```

H. 组合数问题 / Problem

时间限制： 10.0 秒

空间限制： 512 MB

【题目背景】

友情提示：本题篇幅较长。

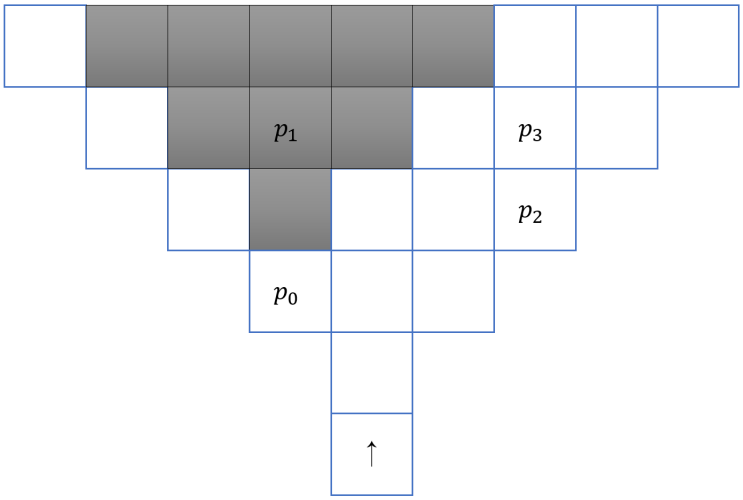
【题目描述】

众所周知，小葱同学擅长计算，尤其擅长计算组合数。

于是现在我们有了一个 $N \times M$ 的棋盘，棋盘上有些地方会有障碍物的存在。在棋盘上有 K 枚棋子，每枚棋子要么属于红队要么属于蓝队。此外，每枚棋子有两个属性值，第 i 枚棋子的两个属性记为 a_i, b_i 并且保证 $a_i \geq b_i$。既然有了两支队伍那么小葱同学就要来玩游戏，这个游戏一共有 R 个回合。每个回合内，每枚棋子会按照如下顺序进行游戏：

0. 回合开始，记当前是第 H 回合。若现在为游戏的开始时刻，则 $H = 1$。
1. 对于所有能够移动的棋子，沿着自己当前的方向移动一格。每枚棋子的朝向一定是上下左右中的一种。如果棋子发现自己不能移动，即前方格子为棋盘外部或者障碍物，那么则不向前移动，而是将自己的方向旋转一百八十度，即进行向后转的操作（注意此时转了之后棋子这回合内并不会进行移动）。
2. 对于所有棋子，每个棋子都拥有且仅拥有一种技能。此时所有棋子按照编号从小到大依次释放技能，如果不能释放技能则跳过。技能总共分为以下几种：
    1. 重锤火花（toolihai）：以自身为中心，制造出绚丽的烟花，令人赏心悦目，无任何效果。
    2. 面子之王（faceking）：该棋子充满了面子，所有棋子都要听其号令，使得所有棋子的方向强制向左（逆时针）转九十度。该技能的效果持续时间为永久。
    3. 头发掉光（onepunch）：头发掉光之后，王队长变得更强了，使该枚棋子的属性大幅度提升。即若当前棋子 i 的属性为 (a_i, b_i)，属性的提升值为 v，那么该棋子的属性则变为 $(a_i + v, b_i)$。该技能的效果能持续 l 个回合，若当前回合为 H，则该技能效果会在第 $H + l$ 回合开始的时候消失。若在技能消失前棋子 i 的属性为 (a_i, b_i)，则技能效果消失后其属性会变为 $(\max(a_i - v, 0), \min(b_i, \max(a_i - v, 0)))$。注意再该技能效果消失前该技能可能被再次释放，则多次释放的技能效果叠加，但技能效果消失的时间分开计算。（详见样例第二组数据）

4. 拉比哩比（rabiribi）：挥动真正的重锤，制造火花，击退前方扇形区域的所有敌方单位。该技能所能影响到的范围为前方大小为 v 的扇形区域，所谓扇形区域可见下图：



从上图我们可以看出，在图的最下方有正在释放技能的棋子，其技能朝向为上，所以其技能方向也为上。则上方的所有区域则为其技能所影响到的扇形区域，该扇形区域大小为 5，即 $v = 5$。注意到棋盘上某些位置会有障碍物，障碍物的存在会影响到该技能的影响范围。如果一个障碍物存在于技能范围之内，那么以障碍物开始的处于技能范围内的扇形区域也都不会受到技能的影响。如上图可见，所有黑色格子的最下方格子即为障碍物的位置，所有黑色格子即为受到障碍物影响所不能被技能效果影响的格子。

另一方面，击退效果的定义为将受到技能影响效果的棋子沿着技能释放方向强制移动一格，即从图中 p_2 被击退到 p_3 的位置。如果被击退方向是障碍物或者棋盘边界，则按一下规则进行：

- 如果下一位置为障碍物，则被击退到该障碍物后面，即从图中的 p_0 击退到 p_1。
- 如果下一位置为棋盘外部，如当前在第一行并且要被击退到第零行，则被击退到第 N 行对应的同一列的位置。

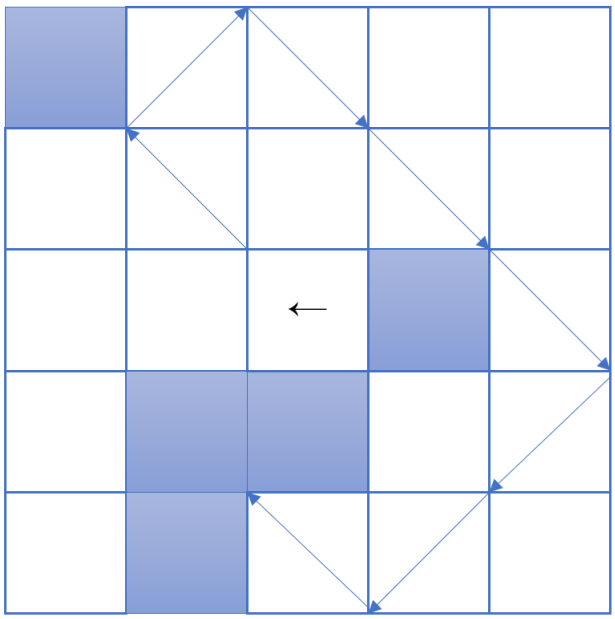
重复以上过程直到被击退到一个不是障碍物和棋盘外部的位置，则该击退过程结束。（详见样例第一组数据）

5. 退学花火（firework）：退学之后，有了更多的时间来挖坑。使得上一个释放重锤火花的棋子在接下来 l 回合内无法移动。如果不存在这样的棋子，则使自己在接下来 l 回合内无法移动。

6. 救乌干达（viuganda）：使用 vim，拯救阵亡的棋子，复活距离自身最远的阵亡的己方棋子。若己方没有阵亡棋子，则复活对方距离自身最近的阵亡棋子。如果对方也没有阵亡棋子，则什么也不做。我们这里使用的距离为曼

哈顿距离，即若第 i 枚棋子处于位置 (x_i, y_i)，第 j 枚棋子处于位置 (x_j, y_j)，那么此时两枚棋子的距离为 $|x_i - x_j| + |y_i - y_j|$。如果同时有多枚棋子距离当前棋子的曼哈顿距离一样，那么我们选择其中编号最大的棋子即可。棋子被复活后如果当前回合可以释放技能并且仍然没有被跳过，则会释放技能。

7. 降维打击（2dsaigao）：利用超时空武器，向自己右前方发射动感光波，使得某个区域内的单位不能释放技能。该动感光波发射之后，遇到棋盘边缘或者障碍物则会发生方向改变的现象，如下图所示：



中间格子为当前棋子所处的位置，其方向向左。每当该光波会进入一个有障碍物的格子或者棋盘边缘的时候，该光波就会改变自己的方向顺时针旋转 90°。该光波每从一个格子的中心到达另一个格子的中心时我们称该光波走了 1 的距离。当光波走了 o 的距离之后，光波就会以当前格子为中心发生爆炸，使得所有距离当前格子曼哈顿距离不超过 v 的棋子在接下来 l 回合的时间内都不能释放技能，即从现在到第 $H + l$ 回合开始的这段时间内都不能释放技能。（详见样例第一组数据）

8. 咕咕咕咕（gugugugu）：做人，绝不能普普通通的鸽了别人；要鸽，必须带着大家一起鸽。该技能效果为使得己方所有当前回合还没有释放技能的棋子全部跳过技能释放的环节，即使得己方还未释放技能的棋子当前回合不能释放技能。

9. 反向传播（backward）：让时间暂停，让游戏倒流，让自身回到过去，回到之前的某个回合。若当前为第 H 回合，该技能效果是回到 l 回合前，则当前棋子 i 会回到第 $H - l$ 回合。所谓回到第 $H - l$ 回合，即当前棋子的一下

所有值都会变回第 $H-l$ 回合开始的时候的值：

- 位置及方向。
- 属性。
- 是否存活以及复活时间。
- 受到的技能影响的效果以及剩余时间。

注意此时棋子 i 的技能刷新时间并不会回到第 $H-l$ 回合的时间。（详见样例第三组数据）（技能刷新时间的规则会在下面说明）

10. 葱之礼赞（hupraise）：葱的到来，使得今天变为了一个神圣的日子，要将红蓝双方的棋子进行配对。每个红方棋子可以和任何一个蓝方棋子进行配对，并且每个棋子都最多只能和对面一个棋子进行配对。第 i 枚棋子和第 j 枚棋子能够配对的条件为两个棋子属于不同阵营并且 $a_i \oplus a_j$ 是 3 的倍数（该符号为异或）。现在每当该技能发动时，你需要求出最多能够配多少对。

技能刷新规则：对于所有技能，当前为第 H 回合，如果第 i 枚棋子释放了技能，那么在第 $H+z_i$ 回合的时候该技能才回复完毕，才能再次释放技能。如果第 i 枚棋子在当前回合中了一个持续时间为 l 回合的技能，那么该技能的影响将在第 $H+l$ 回合开始的时候消除。对于所有棋子，若其首轮技能回复时间为 f_i，技能回复时间为 z_i，则若没有其他棋子的影响，其第一次将在第 f_i 回合释放技能，之后每 z_i 回合再次释放技能。

3. 对于棋盘上所有格子，如果一个格子内既有红队棋子又有蓝队棋子，那么此时这个格子内的棋子便会发生交战。交战的规则如下：
我们规定第 i 枚棋子的战斗力为 $\binom{a_i}{b_i}$（该式子等价于 $C_{a_i}^{b_i}$）。在交战时，每次由红蓝双方派出己方在这个格子战力最大的棋子进行交战。假设红方派出第 i 枚棋子，蓝方派出第 j 枚棋子。如果红方蓝方两枚棋子战斗力一样，那么派出的两枚棋子均阵亡。若战斗力不一样，假设第 i 枚棋子战力更高，那么第 j 枚棋子阵亡，同时第 i 枚棋子的两个属性变为 $\max(a_i-a_j,0),\max(b_i-a_j,0)$。不断重复以上操作直到一方棋子全部阵亡。若第 i 枚棋子在当前回合即第 H 回合阵亡，那么第 i 枚棋子会在第 $H+r_i$ 回合开始的时候复活。第 i 枚棋子复活之后将其两个属性将变为游戏开始时候的属性值，并且移除所有当前棋子身上的技能效果，其移动方向保持阵亡前的移动方向。棋子阵亡期间其技能正常回复，但是不能移动和释放技能，并且不受到其他任何技能的影响。

现在小葱告诉你所有与这游戏相关的东西，希望你按照游戏规则来进行游戏，并输出游戏结束之后每个棋子的位置。

【输入格式】

从标准输入读入数据。

输入包含多组数据，第一行一个整数 T 表示数据组数。接下来依次描述每组数据，对于每组数据：

- 第一行四个整数 N, M, K, R。
- 接下来 N 行每行 M 个数。对于第 i 行第 j 列的数，如果等于 0 则代表第 i 行第 j 列的格子为普通格子，否则为障碍物。
- 接下来 K 行，每行一开始有九个整数 $x_i, y_i, a_i, b_i, c_i, d_i, f_i, z_i, r_i$。代表第 i 枚棋子一开始的位置位于第 x_i 行第 y_i 列，其属性值为 a_i, b_i 并且所属的阵营为 c_i。如果 $c_i = 0$ 则为红方否则为蓝方。棋子 i 一开始的朝向为 d_i，$d_i = 0, 1, 2, 3$ 分别表示上下左右四个方向。f_i 为技能首轮回复时间，z_i 为技能回复时间，即第 i 枚棋子若在没有受到其他棋子影响的情况下，第一次将在第 f_i 回合释放技能，之后每 z_i 回合再次释放技能。r_i 为其复活时间。接下来一个字符串 s_i 为其技能的英文名字，已在每个技能后面注明了其对应的英文名字。对于不同的技能种类，会在当前行增加一定的输入值，具体如下：
  1. 重锤火花（toolihai），无增加的输入。
  2. 面子之王（faceking），无增加的输入。
  3. 头发掉光（onepunch），增加输入 v_i, l_i，代表该技能的增益效果和持续时间。
  4. 拉比哩比（rabiribi），增加输入 v_i 代表技能的范围。
  5. 平安花火（firework），增加输入 l_i 代表技能的影响时间。
  6. 救乌干达（viuganda），无增加的输入。
  7. 降维打击（2dsaigao），增加输入 y_i, v_i, l_i，代表光波能移动的距离、光波爆炸后能影响的范围以及技能的影响持续时间。
  8. 咕咕咕咕（gugugugu），无增加的输入。
  9. 反向传播（backward），增加输入 l_i，表示回到 l_i 回合之前，数据保证不会回到游戏开始之前。
  10. 葱之礼赞（hupraise），无增加的输入。

【输出格式】

输出到标准输出。

对于每组数据：

- 每当技能 10 发动时，输出一行一个整数。
- 在游戏结束之后，输出 K 行，每行两个整数代表第 i 枚棋子现在处于哪一行哪一列。

【样例 1 输入】

```
4
2 4 4 2
0 0 0 0
0 0 0 1
1 2 1 1 0 2 1 10 1 2dsaigao 1 0 20
2 1 1 1 0 0 1 10 1 faceking
2 1 1 1 0 3 1 10 1 rabiribi 1
1 3 1 1 0 1 1 10 1 toolihai
1 5 4 4
0 0 0 0 0
1 1 2 2 0 3 1 1 100 onepunch 1 2
1 2 1 1 1 0 1 1 100 toolihai
1 3 1 1 1 0 1 1 100 toolihai
1 4 2 1 1 0 1 1 100 toolihai
1 4 1 4
0 0 0 0
1 1 1 1 0 3 3 2 2 backward 2
10 10 10 20
0 0 0 1 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 1
0 1 0 0 1 0 0 1 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 1
0 1 0 0 0 0 0 0 0 0
1 0 0 0 1 0 1 0 1 1
0 0 0 0 0 0 0 0 0 1
0 0 1 0 1 1 0 1 0 1
1 0 0 0 0 0 0 0 0 0
10 9 8 0 0 1 6 7 2 viuganda
5 3 10 0 1 1 6 3 7 toolihai
2 7 6 3 0 3 7 3 1 faceking
5 7 10 9 1 3 9 8 8 rabiribi 10
6 10 1 1 0 1 11 7 7 backward 9
2 3 8 5 0 3 2 9 4 onepunch 6 1
1 7 8 2 0 3 11 2 5 hupraise
```

```
9 9 4 3 1 3 12 4 9 gugugugu
2 1 10 6 1 2 6 6 2 2dsaigao 9 9 9
10 10 10 5 1 1 12 4 8 firework 4
```

【样例 1 输出】

```
1 1
1 1
2 3
2 1
1 4
1 2
1 3
1 4
1 2
5
10 6
2 5
1 5
5 8
6 10
1 7
1 5
8 9
4 9
8 1
```

【样例 1 解释】

对于第一组数据，在左上方释放技能 7 时，是朝着角落里射出的动感光波，所以碰到角落返回回射出点时，此时距离为 1，正好原地爆炸，使得技能 2 无法放出。而右下方是在释放技能 4 时，将对方击穿了地图的例子。

对于第二组数据，第一枚棋子每次向右走一步就会释放一次技能。第一回合走到 $(1,2)$，释放技能后属性变为 $(3,2)$，和对方交战后属性变为 $(2,1)$。第二回合走到 $(1,3)$，释放技能后属性变为 $(3,1)$，和对方交战后属性变为 $(3,0)$。第三回合走到 $(1,4)$，由于之前第一回合释放的技能效果消失，属性变回 $(2,0)$，再重新释放技能变为 $(3,0)$，然后被对手痛殴致死。所以最后停留在了 $(1,4)$。

对于第三组数据，该棋子每走两次就会释放技能跳回两回合前的状态，不断循环这

样的过程。

【数据规模】

保证 $1 \le T \le 20$，$1 \le N, M \le 100$，$1 \le K \le 200$，$1 \le R \le 1000$。

保证所有的附加输入除了技能 $3, 5, 7$ 的影响时间大于 0 以外，剩下均大于等于 0。

保证棋子一开始不在障碍物上，一开始以及增幅的属性值均不超过 1000（但不保证增幅后的属性超过 1000）

保证技能的首轮回复时间、之后的回复时间以及复活时间均大于 0。

I. 城市地铁规划 / City

时间限制： 1.0 秒

空间限制： 512 MB

【题目描述】

经过选拔，Kiana 成为了可乐城的市长，为了兑现选举承诺，她决定在可乐城的 n 个重要地标之间修建地铁。

可乐城的交通状况并不复杂，在任意两个地标之间修建一条地铁轨道都是可行的，但是地铁轨道并不是越多越好，如果有太多地铁从一个地标处经过，该地标的拥堵程度将大幅增加。为此，Kiana 决定给每个地标一个便利度来衡量拥堵程度，如果有 d 条地铁轨道经过了某个地标，那么该地标的便利度为 $f(d) \mod 59393$，其中 $f(x) = \sum_{i=0}^{k} a_i x^i$ 是 Kiana 指定的一个 k 次多项式。

因为修建地铁有一定的成本，所以 Kiana 希望新建的地铁轨道尽可能少，但任意两座地标之间都需要能通过地铁相互到达。Kiana 想知道在给定的条件下，什么样的修建方案可以使得地标的便利度之和最大。由于她不会做，所以希望你来告诉她答案。

【输入格式】

从标准输入读入数据。

输入第一行包含两个正整数 n 和 k，分别表示可乐城的地标总数和 Kiana 指定的多项式次数，地标的编号依次为 1 至 n，数据保证 $n \le 3000$ 且 $k \le 10$。

输入第二行包含 $k+1$ 个非负整数 $a_0 \sim a_k$，即 Kiana 指定的多项式的系数，数据保证所有的 $a_i \le 50$。

【输出格式】

输出到标准输出。

输出由若干行组成，第一行包含两个用空格隔开的正整数 m 和 S，分别表示你的方案中修建的地铁轨道数量与最终的便利度之和。

接下来 m 行，每行包含两个用空格隔开的正整数 u 和 v，表示在第 u 和第 v 个地标之间修建了一条地铁轨道。

本题将采用 Special Judge，如果有多种方案使得地标的便利度之和最大，输出其中任意一种即可。

【样例 1 输入】

```
4 2
0 0 1
```

【样例 1 输出】

```
3 12
1 2
1 3
1 4
```

【样例 2 输入】

```
10 9
10 9 8 7 6 5 4 3 2 1
```

【样例 2 输出】

```
9 177454
4 5
4 6
3 4
3 7
2 3
2 8
1 2
1 9
1 10
```

J. 弗雷兹的玩具商店 / Toyshop

时间限制： 2.0 秒

空间限制： 512 MB

【题目背景】

物是人非事事休，欲语泪先流。

【题目描述】

弗雷兹在 C 市有一个玩具店，店里有 n 种玩具，编号依次为 $1,2,\ldots,n$，编号为 i 的玩具的单价为 c_i 元，一个该玩具提供的愉悦度为 v_i 。

突然有一天，C 市来了 m 个小朋友。据可靠消息，这些小朋友会在一些时刻一起来店里买东西，其中第 i 个小朋友每次都会带 i 元（$1 \le i \le m$ ）。

由于某些玩具特别优秀，所以每次小朋友们都会在特定的编号范围内挑选玩具。

除此之外，由于小朋友们在一年前的清华校赛中就愉悦得无法自拔，所以弗雷兹放弃了对他们的治疗，于是小朋友们就可以无限制地购买玩具了。也就是说，对于任意玩具，每个小朋友在每次的购买件数都可以是任意的非负整数。

时代飞速发展，玩具的受欢迎程度和价格也会随着时代的发展而改变。

为了方便你处理这些信息，Yazid 进行了整理，发现这些日子里，弗雷兹的玩具商店里共发生了 Q 个事件。

对于每个事件，都有 3 个基本参数 op, l, r 。其中 op 为 1 至 3 之间的整数，代表了事件的类别：

1. 对于 $op = 1$ 的事件，Yazid 还会给你一个额外参数 d ，表示这是一个**价格调整**事件：将编号在区间 $[l, r]$ 内的玩具的单价 c 全部增加 d 元。为了防止单价超过 m 元导致玩具永远无法被小朋友们购买，弗雷兹会将所有超过 m 的单价减去 m。（保证 d 为不超过 m 的正数）
2. 对于 $op = 2$ 的事件，Yazid 还会给你一个额外参数 b ，表示这是一个**愉悦修正**事件：将编号在区间 $[l, r]$ 内的玩具的愉悦度 v 全部增加 b 。（需要注意这里的 b 可能是负数）
3. 对于 $op = 3$ 的事件，表示**购买**事件：所有的 m 个小朋友来到弗雷兹的玩具商店，在编号范围在 $[l, r]$ 内的玩具中进行随意地购买。

现在，对于每一次的购买事件，你想知道：

1. 所有小朋友所能获得的最大愉悦度之和。
2. 所有小朋友所能获得的最大愉悦度的异或和（异或运算是 xor 运算，即 C++/Java/Python 中的 ^ 运算）。

【输入格式】

从标准输入读入数据。

- 第 1 行 2 个正整数 n, m，分别表示玩具数量、以及小朋友的数量。
- 第 2 行 n 个正整数 $c_1, \ldots, c_n$，依次描述各玩具的单价。
- 第 3 行 n 个正整数 $v_1, \ldots, v_n$，依次描述各玩具的愉悦度。
- 第 4 行 1 个非负整数 Q，表示事件的数量。
- 接下来 Q 行依次描述所有事件，每行描述一个事件。每行的开头是 3 个整数 op, l, r，意义见题目描述。
  - 如果 $op = 1$，接下来跟随 1 个该事件的额外参数（整数）d，意义见题目描述。
  - 如果 $op = 2$，接下来跟随 1 个该事件的额外参数（整数）b，意义见题目描述。
  - 如果 $op = 3$，接下来无任何额外参数。
  - 保证 $1 \le l \le r \le n$。

对于每一行，如果包含多个数，则用单个空格将它们隔开。

【输出格式】

输出到标准输出。

- 对于每个购买事件，输出一行 2 个用空格隔开的整数，依次表示所有小朋友所能获得的最大愉悦度之和、以及所有小朋友所能获得的最大愉悦度的异或和。

【样例 1 输入】

```
4 10
1 6 10 2
100 2333 666 300
7
3 1 4
3 1 3
1 2 4 1
3 1 4
2 2 3 -1000
2 2 3 -600
3 2 4
```

【样例 1 输出】

```
15165 2865
14165 2169
36630 798
4899 1273
```

【样例 1 解释】

对于第 1 个购买事件，各位小朋友（编号从小到大，即从 1 至 10）能够获得的最大愉悦度依次为：$100, 300, 400, 600, 700, 2333, 2433, 2633, 2733, 2933$。

对于第 2 个购买事件，各位小朋友（编号从小到大，即从 1 至 10）能够获得的最大愉悦度依次为：$100, 200, 300, 400, 500, 2333, 2433, 2533, 2633, 2733$。

对于第 3 个购买事件，各位小朋友（编号从小到大，即从 1 至 10）能够获得的最大愉悦度依次为：$666, 1332, 1998, 2664, 3330, 3996, 4662, 5328, 5994, 6660$。

对于第 4 个购买事件，各位小朋友（编号从小到大，即从 1 至 10）能够获得的最大愉悦度依次为：$0, 0, 300, 300, 300, 600, 733, 733, 900, 1033$。

根据这些信息，你将很容易计算出答案。

【数据规模】

保证 $1 \le n \le 200,000$，$1 \le m \le 60$，$0 \le Q \le 30,000$。

保证 $1 \le c_i, d \le m$。

保证 $0 \le v_i \le 10^7$，$|b| \le 10^3$。

【提示】

这个提示本不该有，但善良的出题人还是想提醒你：所有小朋友所能获得的最大愉悦度之和有可能超过 32 位有符号整数的范围。

K. 蛋糕 / Cake

时间限制： 1.0 秒

空间限制： 512 MB

【题目描述】

最近，菲菲学会了做蛋糕，她做了一个 **4D** 的蛋糕送给牛牛，这个蛋糕的大小是：$a \times b \times c \times d$。

蛋糕所有的表面都抹着奶油，牛牛想蛋糕沿着与表面平行的超平面，切成 $1\times1\times1\times1$ 的小块。

现在，牛牛想知道，在这些小块蛋糕中，有 0 个、1 个、2 个.....8 个表面抹着奶油的有分别有多少块。

【输入格式】

从标准输入读入数据。

输入包含多组数据，第一行一个整数 T 表示数据组数。（$T <= 10000$）

接下来依次描述每组数据，对于每组数据：

- 一行 4 个正整数 a、b、c、d。（四个正整数都在 10^6 以内）

【输出格式】

输出到标准输出。

对于每组数据，输出 1 行：

- 一行输出 9 个整数，分别表示有 0 个、1 个、2 个.....8 个表面抹着奶油的小块蛋糕数量，所有值之间用单个空格隔开。为了防止答案过大，你只需要输出答案除以 2148473648 的余数。

【样例 1 输入】

```
5
2 2 2 3
7 7 8 8
9 9 9 9
3 3 5 6
9 7 5 2
```

【样例 1 输出】

```
0 0 0 8 16 0 0 0 0
900 1320 724 176 16 0 0 0 0
```

```
2401 2744 1176 224 16 0 0 0 0
12 62 108 72 16 0 0 0 0
0 210 284 120 16 0 0 0 0
```

【样例 1 解释】

对于第一组数据，把 $2\times2\times2\times3$ 的蛋糕切成小块后，共形成了 24 个小立方体，其中有 16 个小块分别与其他 4 个小块相邻，因此它们有 4 个面涂有奶油，而另外 8 个小块分别与其他 5 个小块相邻，因此它们有 3 个面涂有奶油。

L. 生生不息 / Lives

时间限制： 1.0 秒

空间限制： 512 MB

【题目描述】

生命游戏是一个经典的零玩家游戏。

游戏在一块 $n \times m$ 的方格棋盘上进行，初始时，棋盘上的一些格子中有生命，另一些格子中没有生命。

在新的一天开始时，如果一个格子周围的 8 个（边界上的格子也许不到 8 个）格子中，在前一天有恰好 2 个或 3 格子生命，则这个格子上的生命可以继续生存，如果周围的格子中有 3 个格子在前一天有生命且这个格子中前一天没有生命，则会在这个格子中诞生新的生命。在其他情况下，盖格子中原有的生命则会死去且不会产生新的生命。

如果在某一天，棋盘上所有的格子里都没有生命，显然，在接下来的时间里，所有的格子中再也不会有生命了，我们就说这些生命灭绝了。

现在，牛牛希望让菲菲来决定游戏开始时棋盘上的每个格子中是否有生命。

而他想知道，在游戏开始时，菲菲有多少种不同的方法使得棋盘上的生命永远也不会灭绝。

【输入格式】

从标准输入读入数据。

输入包含多组数据，第一行一个整数 T 表示数据组数。（$T <= 30$）

接下来依次描述每组数据，对于每组数据：

- 一行 2 个正整数 n、m。（$n\ m <= 5$）

【输出格式】

输出到标准输出。

对于每组数据，输出 1 行：

- 一行输出 1 个整数，表示问题的答案。

【样例 1 输入】

```
3
2 2
3 3
```

【样例 1 输出】

```
5
150
150
```

M. 为最美的那位献上花朵 / M

时间限制： 1.0 秒

空间限制： 512 MB

【题目描述】

人生是怎么样的呢？我们是怎样活着的呢？

无论是谁，都是从黑暗中降生，最后再回归于黑暗。几年，几十年，短暂而漫长的时光。形形色色的人，形形色色的事，如过眼云烟，转瞬即逝。过眼繁花终有尽时，有些东西还能弥留于你的记忆之中，但更多的，早已无可寻觅。

但是，有个人，你不会忘记。

在你降生于黑暗之中时，那个人就一直陪在你身边。当你突破那黑暗，第一次见到世界的光明的时候，那个人也在你身边。

第一次的啼哭，第一次的说话，第一次的走路，第一次的奔跑。

每一次的成长，是你自己的一次飞跃。而对那个人来说，则是最纯真的喜悦和感动。

在你不知道的时候，在你不知道的地方，那个人倾尽自己的一切，为你创造最好的一切，保护你的一切，为你指引走向未来的桥梁。

那个人每天都在期盼着，期盼着你能早点长大，变得拥有保护你自己的力量。

那个人每天都在担心着，担心着你与那个人分别那天，终究还是会来到。

你终究会一直成长，你终究会拥有自立自强的能力。那个人，也不能再像以前一样，保护你。

有一天，你将走向更宽阔的未来，你会拥有和以前完全不一样的一切。而那个人，已经不再能够保护你了。

而你和那个人之间能够拥有的时间，也就越来越短了。

五年，十年，二十年，你一天一天长大。

五年，十年，二十年，那个人一天天老去。

那个人已经不能保护你了，那么现在，该你，去保护那个人了。

也许那个人不能永远在你身边，也许那个人终将离你而去。

但你永远不会忘记那个人，因为那个人，始终在你身边。

今天，是那个人的节日。

今天，就是母亲节。

而今天，你应该给她，献上什么样的花朵呢？

【输入格式】

从标准输入读入数据。

输入不为空，但输入的东西对这题并没有任何用。

【输出格式】

输出到标准输出。

答案是一种花的名字，输出一个长度不超过 1000 的字符串，代表花的名字的拼音或者英文名字。例如如果答案为樱花，那么 yinghua、cerasus 或者 cherry 都会被认为是正确的，但是 sakura、yinhua、daanshiyinghua 都不是一个正确的答案。输出不区分大小写，即你可以将任意小写字母替换为大写字母，答案仍然正确。

【提示】

如果你不知道该献上的花朵是什么，请不需要慌乱。

我们把它作为彩蛋藏在了这份试题中，好好找一找吧！