第三届河北省大学生程序设计竞赛

试

题

册

河北·秦皇岛

2019.5.12

Problem A. Battle of Balls

Time limit: 1000ms
Memory limit: 65536KB

Description

Now that you have a ball of radius r, you want it to go from the bottom boundary of the map to the top boundary of the map (starting and ending points can be considered outside the map, entering the map from the bottom boundary and leaving the map from the top boundary).

But there are a lot of little spikes in the map, and we can think of them as points. Your ball can't touch any of the points and the left or the right boundary of the map, even if the edge of the ball can't touch them. The area of the map can be seen as 100×100, with the lower left corner(0,0)and the upper right corner(100,100). The bottom boundary is $y = 0$, and the top boundary is $y = 100$.

Input

The first line contains an integer T, representing the number of data sets.

After that, there were T sets of data. In each set:

The first line contains an integer n ,which represents the number of small spikes, and a floating point number r, which is your ball's radius.

The next n lines, each line containing two floating point numbers x, y, representing the coordinates of the spikes.

$1 \le T \le 100,\ 0 \le n \le 1000\ ,\ 0 \le r \le 1000,\ 0 \le x, y \le 100$

Output

For each set of input, if the ball can reach the top boundary of the map, output Yes ; Otherwise, output No.

Sample input	Sample output

1
1 25.0
50.0 50.0
	No

Problem B. Icebound and Sequence

Time limit: 1000ms
Memory limit: 65536KB

Description

Icebound hates math. But Imp loves math. One day, Imp gave icebound a problem. The problem is as follows.

$$S = (\sum_{i=1}^{n} q^{i}) \ mod \ p$$

For given q, n, p, you need to help icebound to calculate the value of S.

Input

The first line contains an integer T, denoting the number of test cases.

The next T lines, each line contains three integers q, n, p, separated by spaces.

$$1 \le T \le 100,\ 1 \le n, q, p \le 10^9$$

Output

For each test case, you need to output a single line with the integer S.

Sample input	Sample output

2
2 3 100
511 4 520
	14
	184

Problem C. 分治

Time limit: 1000ms
Memory limit: 65536KB

Description

你是 DEEP 国的大军师，辅佐一个非常有野心的国王，这位国王非常有野心，他计划攻占 n 个国家。在地图上，这些国家排成一行。

探子已经查明，当攻打一个国家 i 时，为了防止国家间的联合对抗，需要给该国家周围，所有未被攻占的国家支付$cost_i$ 个金币，即对于国家 i，它左侧第一个已被攻打的国家为 l，右侧第一个已被攻打的国家为 r，则他需要给$[l+1,i-1]$ 和 $[i+1,r-1]$ 之间的国家支付金币。如果 l 不存在，则需要给 $[1,i-1]$ 之间的所有国家支付金币；若 r 不存在，则需要给 $[i+1,n]$ 之间的所有国家支付金币。

现在，你的下属已经给你提供了每个国家需要支付金币的数量。为了满足国王的野心，你需要计算出攻占完所有国家需要的最小花费。

Input

第一行是一个整数 T，代表接下来有T组数据。

接下来每组数据

第一行有一个整数 n，代表要攻占的国家数目。

第二行共 n 个整数，代表攻占每个国家前，需要支付给其周围未被攻占国家 $cost_i$个金币。

$$1 \le T \le 50，1 \le n \le 100，1 \le cost_i \le 10000$$

Output

对于每组数据输出一行，代表需要支付的最小花费。

Sample input	Sample output

2
1
1
3
1 1 2
	0
	2

Problem D. 榜单

Time limit: 1000ms
Memory limit: 65536KB

Description

给定一场 UCPC 比赛的题目数量和提交列表，请你输出终榜。

榜单的格式见输出和样例

根据比赛规则，榜单有以下要求：

1. 每道题的通过罚时按照分钟计算。每次未通过提交增加 **20** 分钟罚时，保证每个队伍罚时均小于 **10000** 分钟。到比赛结束都没有通过的题目不计入该队伍的罚时。
2. 每队通过后的题目在榜单的题目栏中用 + 显示。如果仅提交一次就通过了，则显示 +。否则显示+k，k为这个队伍对于这道题，第一次通过前，所提交未通过的次数。保证 k 不大于 9。
3. 每队提交但是未通过的题目在榜单的题目栏中用 -k 显示，k为这个队伍对于这道题的总提交次数。保证 k 不大于 9。
4. 每队没有提交的题目需要在榜单的该队的题目栏留空。
5. 由于大家都讨厌 CE，所以状态为 Compile Error 的提交不计入榜单。
6. 众所周知，对于某个队伍来说，在通过某个题目后再次提交该题目，则通过后的提交**不计入榜单**。
7. 如果某个队伍没有提交，或者所有的提交均不计入榜单。则榜单上**不显示该队伍**。

Input

第一行一个数字 n，表示这场比赛有 n 道题目，题目的标号从 A开始。

接下来若干行，每行格式形如：时间 题号 结果 队名，表示一条提交，提交按时间顺序排列。其中，时间形如 HH:MM，并且时间一定小于 05:00。

题号为单独的一个大写字母。

结果属于集合 { Accepted, Wrong Answer, Time Limit Exceeded, Compile Error, Memory Limit Exceeded, Output Limit Exceeded, Runtime Error, Presentation Error }。

队名为一个含有空格、大写和小写字母的字符串，队名长度不超过 50。

输入以一行 GAME OVER! 结尾，表示比赛结束。

数据保证最多有 5000 条提交记录。$n \le 14$

Output

输出的榜单有 $n + 4$ 栏，每栏之间间隔 2 个空格。

$Rank$ 一栏的宽度为 4 个字符，表示该队伍的排名

Who 一栏的宽度为所有显示在榜单上的队伍名字的**最长长度**，表示该队伍的名字

$Solved$ 一栏宽度为 6 个字符，显示每个队伍通过题目的数量

$Penalty$ 一栏宽度为 7 个字符，按要求显示每队罚时

接下来是题目栏，每个题目栏的宽度均为 3 个字符，按要求显示+或-，表示每个队伍通过题目的情况

每一栏的第一行为这一栏的名称，其中，*Who* 需**左对齐**，其他栏需**右对齐**。

题目栏的名称为题目的标号

接下来若干行，按顺序输出每个队伍的信息，每栏的信息需**右对齐**。

队伍按照通过题目数量排名，如果两队通过题目数量相等，罚时少的队伍排名靠前。

注意，如果出现题数和罚时均相等的队伍，则按照队名的**字典序排序**，同时*Rank* 一栏的值需相等。第一个与他们排名不相等的队伍的排名可以是绝对排名或相对排名。例如前 3 个队伍的排名分别为 1，2，2，则第 4 个队伍的绝对排名为 4，相对排名为 3，你的程序只需要按照一种方式输出即可，但是你需要保证你的程序对于所有输入均按照一种方式输出。

Sample input

```
4
00:01 B Wrong Answer University of Deep Dark Fantasy
00:01 B Accepted University of Deep Dark Fantasy
00:01 C Accepted University of Deep Dark Fantasy
00:01 D Accepted University of Deep Dark Fantasy
00:11 A Accepted Deep Dark Institude of Fantasy
00:13 C Wrong Answer Banana University
01:01 C Wrong Answer Banana University
01:11 C Wrong Answer Banana University
02:01 C Runtime Error Deep Dark Institude of Fantasy
02:10 C Accepted Deep Dark Institude of Fantasy
02:30 A Accepted University of Deep Dark Fantasy
02:50 D Accepted Bon Sha Ka La Ka Higher School of Economics
02:51 C Accepted Bon Sha Ka La Ka Higher School of Economics
02:52 B Accepted Bon Sha Ka La Ka Higher School of Economics
02:53 A Accepted Bon Sha Ka La Ka Higher School of Economics
02:55 A Runtime Error University Van Billy
02:59 B Compile Error University Van Banana
GAME OVER!
```

Sample output

```
Rank  Who                                          Solved  Penalty    A    B    C    D
   1              University of Deep Dark Fantasy       4      173    +   +1    +    +
   2  Bon Sha Ka La Ka Higher School of Economics       4      686    +    +    +    +
   3               Deep Dark Institude of Fantasy       2      161    +        +1
   4                            Banana University       0        0             -3
   4                         University Van Billy       0        0   -1
```

Problem E. Paper Plane Fly Away

Time limit: 1000ms
Memory limit: 65536KB

Description

There are n boys, indexed from 1 to n, and n girls indexed from $n+1$ to $2n$.

One day, they have a party together. The girls are seated in the first row, and the boys sit in the second row. They take their seat in such a way, that a boy sit just behind a girl, and boy indexed 1 sit on the leftmost chair, 2 sit on the second, etc.

Each boy has a girl he likes, and he may make contact to his sweetheart, by writing down what he wants to tell her on a piece of paper, and makes it a paper plane, which he will throw directly at her.

You may assume that the plane will go in a straight line. But, the trouble occurs, when two paper plane collide in the air and drop halfway. Obviously, this will be extremely awkward. So each boy wants to know, if he throws his paper plane, how many paper planes have the potential to collide with it halfway. It's guaranteed that each girl is the sweetheart of exactly one boy.

Input

The first line contains a single integer n. Then n lines follow, the i-th of which contains two integers, the index of the girl seated in front of the i-th boy and the girl he likes.

$$1 \le n \le 10^5$$

Output

Output n lines, the i-th of which contains one integer, the number of planes that may collide with i-th boy's plane.

Sample input	Sample output

5
7 9
6 6
8 10
9 7
10 8
	3
	2
	2
	3
	2

Problem F. Take Apples

Time limit: 1000ms
Memory limit: 65536KB

Description

Bob and Alice have three piles of apples. There are M, N and N apples in each pile, respectively.

Clever Alice and idle Bob play the game. Alice and Bob take turns to take the apples from the game. The rules are as follows:

1. Choose a pile and take no more than S apples from it;

2. Or you can take the same number (can be larger than S) of apples from three piles at a time. If there is a pile with no apples, the rule cannot be used to take apples.

When the game starts, Alice takes the apple first. And they both take the optimal strategy.

The one who gets the last apple will win.

For given M, N, S, who will win, Alice or Bob?

Input

The input contains multiple sets of data

For each test case, each line contains three integers S, M, N, separated by spaces.

$$1 \le S, M, N \le 10^9$$

Output

For each test case, outputs a single line with the answer.

If Alice wins, output "Alice"; If Bob wins, output "Bob" (without quotes).

Sample input	Sample output
1 5 3 4 4 3	Alice Alice

Problem G. 点我

Time limit: 1000ms
Memory limit: 65536KB

Description

X 腿与队友到河北省来参加 2019 河北省大学生程序设计竞赛，然而这场比赛的题目难度实在是太高了。比赛开始一个小时后，X 腿仍然没有做出一个题。这时候，X 腿惊讶的发现电脑屏幕上出现了一个神奇按钮，按钮上写着**”点我签到“**。X 腿非常兴奋，于是他点击了这个按钮，只见屏幕上的题目状态由“**未提交**”转变成了“**答案错误**”。他又点击了一下这个按钮，题目状态由“**答案错误**”变成了“**通过**”！当题目状态为“**通过**”时，我们认为 X 腿签到成功了。

通过多次实验，X 腿总结出以下经验：

1. 当题目状态为”**未提交**“时，点击按钮后题目状态将变为”**答案错误**“。
2. 当题目状态为”**答案错误**“时，点击按钮后题目状态将变为”**通过**“。
3. 当题目状态为”**通过**“时，点击按钮后题目状态将变为”**答案错误**“。

现在，已知初始的题目状态为”**未提交**“。由于 X 腿过于兴奋，点击了 n 次按钮。请问 X 腿签到成功了吗？

Input

一行一个正整数 n，代表 X 腿点击了 n 次按钮。

$$0 \leq n \leq 100000$$

Output

输出一行。如果 X 腿签到成功，请你输出**“qiandaochenggong”**；否则输出**"qiandaoshibai”**。（输出均不含引号)

Sample input	Sample output
3	qiandaoshibai

Problem H. 天神的密码

Time limit: 1000ms
Memory limit: 65536KB

Description

2018 年，icebound 打开了神殿。而在 2019 年，icebound 正在试图破解天神的密码，以期获得天神的力量。

Icebound 发现，想要得到神的密码，必须先要完成一个祭祀仪式。在这个祭祀仪式上，我们首先会追随神的指引，得到两个正整数 N和 K。随后，我们令 $X = N^K$，得到天神喜欢的数字X。

利用 X，我们进行以下方式得到天神最后的密码：

步骤 1 将 X每个数位上的数字相加得到 Y。

步骤 2 令 $X = Y$

步骤 3 反复执行 步骤 1，直到 X只有一个数位时停止，即 $1 \leq X \leq 9$。此时的 X即为天神的密码。

比如：当 $N = 11, K = 2$ 时，首先我们得到 $X = N^K = 11^2 = 121$。然后我们把 X 的各个数位上的数相加，即 $Y = 1 + 2 + 1 = 4$。此时 $X = Y = 4$，X 仅有一个数位了，所以我们停止操作，得到天神的密码为4。

Icebound 许诺，如果他获得了天神的力量，一定保你荣华富贵，全家幸福，还会另外送你一块金牌。所以，请你帮助他计算天神的密码。

Input

首先第一行一个整数 T ，代表数据组数。

随后 T 行，每行两个数 N, K ，用空格隔开。

$$1 \leq T \leq 20，1 \leq N \leq 10^9，1 \leq K \leq 2$$

Output

一行一个整数 X，表示天神的密码。

Sample input	Sample output

2
11 2
100 1
	4
	1

Problem I. Twinkle

Time limit: 1000ms
Memory limit: 262144KB

Description

The Cartesian coordinate system is set in the sky. There you can see n stars, the i-th has coordinates (x_i, y_i), a maximum brightness c, equal for all stars, and an initial brightness $s_i (0 \leq s_i \leq c)$.

Over time the stars twinkle. At moment 0 the i-th star has brightness s_i. Let at moment t a star has brightness x. Then at moment $(t+1)$ this star will have brightness $x+1$, if $x+1 \leq c$, and 0 otherwise. You want to look at the sky q times. In the i-th time you will look at the moment t_i and you will see a rectangle with sides parallel to the coordinate axes, the lower left corner has coordinates $(x1_i, y1_i)$ and the upper right — $(x2_i, y2_i)$. For each view, you want to know the total brightness of the stars lying in the viewed rectangle.

A star lies in a rectangle if it lies on its border or lies strictly inside it.

Input

The first line contains three integers n, q, c — the number of the stars, the number of the views and the maximum brightness of the stars.

The next n lines contain the stars description. The i-th from these lines contains three integers x_i, y_i, s_i— the coordinates of i-th star and its initial brightness.

The next q lines contain the views description. The i-th from these lines contains five integers $t_i, x1_i, y1_i, x2_i, y2_i$ — the moment of the i-th view and the coordinates of the viewed rectangle.

$$1 \leq n, q \leq 5 \cdot 10^4, 1 \leq c \leq 10^9, 0 \leq s_i, t_i \leq c$$

For all coordinates, $1 \leq x_i, y_i \leq n$

Output

For each view print the total brightness of the viewed stars.

Sample input	Sample output

3 3 5
1 1 2
2 3 0
3 3 1
0 1 1 3 3
1 2 2 3 3
2 2 1 3 3
	3
	3
	5

Problem J. 舔狗

Time limit: 1000ms
Memory limit: 262144KB

Description

“舔狗舔狗， 舔到最后， 一无所有。”

有 n 只舔狗，每只舔狗的心中都有自己朝思暮想的一位。

每个人虽然受到了一万次拒绝，还毅然第一万零一次鼓起勇气。

作为一个不食人间烟火的算法设计师，你早已看破红尘。但是，人世间的苦难仍让你挂念。看到众生在单恋中苦苦坚持，你决定普度众生，给大家找到一个最好的结局，让一无所有的舔狗尽量地少，让每个人都尽量能和自己喜欢的或喜欢自己的人修成正果。

也就是说，你需要给这 n 只舔狗配对，对于舔狗 i，他可以和他朝思暮想的人 a_i 配对。另外，喜欢 i 的其他舔狗也可以和他配对。你需要让没有被配对的舔狗尽量少。

Input

第一行一个 n，表示舔狗个数。

第二行 n 个数字，第 i 个数字表示第 i只舔狗的朝思暮想的一位的编号 a_i。

$$2 \le n \le 10^6$$

Output

第一行一个数字，表示一无所有的舔狗的最小数量。

Sample input	Sample output
10 3 1 8 6 10 1 4 1 6 1	0

Problem K. 河北美食

Time limit: 1000ms
Memory limit: 65536KB

Description

不知不觉当中，河北成为了一些人心中的“美食荒漠”，除了驴肉火烧，大抵想不起什么河北的美食了。大概是京津太过闪耀，盖过了冀菜的光芒。其实河北并不是美食荒漠，像邯郸的豆沫，石家庄的缸炉烧饼，唐山的酥糖，秦皇岛的皮皮虾，总能勾起心中最美好的回忆。

Icebound 最喜欢吃河北菜，于是他想要大厨做一桌河北菜宴请宾客。icebound 购买了一些食材，并且制订了宴会的菜单。但是他并不知道这些食材是否足够，所以希望你写一个程序帮助他。

Icebound 将会给出每种食材的名称和数量，以及完整的菜单。菜单将包含每种菜品所需的食材及数量。菜单上的每道菜只需制作一次。

Input

第一行给出两个整数 n,m，分别代表食材种类和菜品数量。

第二到第 $n+1$ 行，每行一个由小写字母组成的字符串 s_i 和一个数字 a_i，表示这种食材的名称和数量。

接下来 m 行，每行首先有一个整数 k，代表这种菜品所需的食材种类数。

随后将会有 k 个字符串，代表食材名称，每个字符串后跟有一个数字 t_i，用空格隔开，代表需要的食材数量。

$$1 \le n,m \le 1000，1 \le k \le 10 \text{ 且 } k \le n$$

$$1 \le a_i,t_i \le 10^9，1 \le |s_i| \le 20$$

保证输入合法，食材名称不会相同，且菜谱中不会有未出现的食材。

Output

如果食材足够将菜单上的所有菜品全部制作一遍，请输出一行 “YES”，并且**按照输入顺序**输出剩下的食材以及对应的数量，每行一个食材，用空格将食材和其数量隔开。**如果某种食材全部被用完，则不输出该食材。**

如果不能，输出一行 “NO”。

Sample input	Sample output
5 3 water 100 flour 20 cabbage 71 pork 12 bean 5 2 water 20 flour 5 3 water 70 cabbage 54 pork 10 5 water 1 flour 1 cabbage 1 pork 2 bean 1	YES water 9 flour 14 cabbage 16 bean 4

Problem L. Smart Robot

Time limit: 1000ms
Memory limit: 137102KB

Description

Icebound dreams of being a **Legendary grandmaster**. So he built a smart robot to help him.

The robot works on a $N * N$ matrix. All the numbers in the matrix are non-negative integers less than 10. The robot can move in the matrix. If it's in the (x, y) position of the matrix, it can move to $(x+1, y), (x, y+1), (x-1, y), (x, y-1)$, which are the adjacent positions of the robot. But the robot has to operate inside the matrix. It can't get out of the matrix.

The robot can start at any position in the matrix, take any step, and it can stop at any position in the matrix. We connect the numbers in the robot's path in order to get a magic number. For example, if the robot takes 3 steps and passes through 3 numbers of the matrix, $0, 7$ and 8, the magic number is 78. **All the magic numbers are non-negative integers, and the number generated by the robot does not contain leading 0.**

Through the robot, icebound got a lot of magic numbers. Now he wants to know what is **the smallest magic number he can't get.**

Input

The first line is an integer N, denoting the length and width of the matrix.

The next N lines, N numbers per line, separated by spaces, represent this matrix. Ensure that each number in the matrix is less than 10 and greater than or equal to zero.

The numbers in the matrix are randomly generated.

$$1 \leq N \leq 50$$

Output

Print one integer per line, indicating the smallest magic number that icebound can not get.

Sample input	Sample output

4
1 2 3 4
3 6 7 8
0 1 5 4
9 1 1 1
	17