2016 ACM-ICPC 国际大学生程序设计竞赛陕西省省赛

长安大学

2016 年 05 月 22 日

Problem.A Rui and her cylinders

Time Limit: 1000ms Memory Limit: 256MB

- Description

Rui is magnificently gifted. She wants a cylinder as a Children's Day present.

Her loveliness is like her interest of the shortest path from one point to another on the surface of cylinder.

- Input

There are several test cases in this problem. The first line of input contains a single integer T (about one thousand) denoting the number of test cases.

For each test case, the first line contains two integers: radius and height ($1 \leq radius \leq 100$, $1 \leq height \leq 100$), denoting the radius and height of the cylinder.

For the next two lines, each line contains three integers: h, a and r ($0 \leq h \leq height,\ 0 \leq a \leq 360,\ 0 \leq r \leq radius$), denoting one point on the surface of cylinder each. The h indicates a circle on the surface of cylinder which apart h from the bottom. And the polar angle a and radius r indicates the position of the point on the circle. In the other words, if the cylinder is $(0, 0, 0)$ to $(0, 0, height)$ on the 3D grid coordinate, the point can be represented as $(r\cos a, r\sin a, h)$. You may assume that $r \neq radius$ only when $h = 0$ or $h = height$ for each point.

- Output

For each test case, output only one line contains the length of the shortest path on the surface of cylinder. The answer should be rounded to two digits after the decimal point.

- Sample Input

```
2
5 10
10 0 3
5 0 5
90 49
49 312 39
0 52 65
```

- Sample Output

```
Case #1: 7.00
Case #2: 171.02
```

Problem.B Rui and her functions

Time Limit: 10000ms Memory Limit: 256MB

- Description

Rui is magnificently gifted. Why does she not play with me tonight? Oh, she is obsessing about n functions with n quartette of positive coefficients, denoted by a_i, b_i, c_i, d_i $(1 \leq i \leq n)$ respectively.

The i-th function f_i is defined as $f_i(x) = (a_i b_i^x + c_i)\%d_i$. She asked Doc to find the smallest x_i in $[1, m]$ for each function such that $f_i(x_i)$ achieves the minimum of f_i in $[1, m]$. That is say: $f_i(x_i) = \min\{x \in [1, m] \mid f_i(x) = min_{t \in [1,m]}\{f_i(t)\}\}$.

However n is large and Doc told her that possible x_i for each function is unique (and Rui is unique as well), and $x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n$ (and that is as amazing as Rui).

Now she needs to find x_i by herself.

- Input

There are several test cases (no more than 64) and please process till EOF.

The first line in each case contains two integers n and m, $1 \leq n, m \leq 100000$. Each of the following n lines contains four integers a_i, b_i, c_i, d_i respectively, where $0 < a_i, b_i, c_i < d_i \leq 10^9$.

- Output

For each test case, print first the identifier of the test case, then n lines follow. The i-th line contains the number x_i for i-th function f_i.

- Sample Input

```
3 5
373911025 1525760443 652804587 1767005941
120055457 159868670 59429374 196292251
1200581 955324 141748 2705431
```

- Sample Output

```
Case #1:
1
2
4
```

Problem.C Rui and her sequences

Time Limit: 1000ms Memory Limit: 256MB

- Description

Rui is magnificently gifted. She has pioneered sorting sequence, starting in a time when given sequence S wasn't so ordered. The sequence S is constituted by 0, 1 and 2, and her target is the minimum cost for rearranging this sequence with non-descending order. She has two type of operations as follows.

1. Select a continuous subsequence of S, move all 2 to the front and all 0 to the back in the subsequence, with a unit of cost.

2. select two element of S and swap them, with b unit of cost.

- Input

There are several test cases (no more than 70) and please process till EOF.

The first line in each case contains three integers n, a, b where n is the length of S, $1 \leq n \leq 100000$ and $0 \leq a, b \leq 1000$. The second line contains n integers in $\{0, 1, 2\}$ corresponding with the elements in S.

- Output

For each case, output the minimum cost for sorting S in a line.

- Sample Input

```
3 1 1
0 1 2
3 0 1
1 0 1
8 2 3
1 1 0 1 0 1 0 0
```

- Sample Output

```
0
1
9
```

Problem.D Rui and her triangles

Time Limit: 10000ms Memory Limit: 256MB

- Description

Rui is magnificently gifted. Now, she has a tree T with n vertices (numbered 1 through n) and each vertex i has a stick with length a_i placed at it. The tree is rooted at vertex 1. She looks at the subtree T_i of each vertex i and wants to know how many distinct triples of vertices in this subtree are well-placed.

A triple of vertices is well-placed if the three sticks placed at the three vertices of this triple can form a triangle with positive area. Two triples of vertices are distinct if there exists a vertex which belongs to one of these two triples but does not belong to the other one. Note that a single triple is not allowed to contain duplicate vertices, but different triples may have same vertices.

- Input

The first line of the input contains an integer t ($1 \leq t \leq 100$) - the number of test cases.

For each test case, the first line contains a single integer n ($1 \leq n \leq 1000$) - the number of vertices in tree T. The second line contains $n-1$ integers, the i-th integer is between 1 and i inclusively, indicating the parent vertex of vertex $i+1$. The third line contains n integers, the i-th integer a_i indicates the length of the stick placed at vertex i ($1 \leq a_i \leq 1000$).

- Output

For each test case, print first the identifier of the test case, then n lines follow. The i-th line contains the number of distinct triples of vertices in subtree T_i which are well-placed.

- Sample Input

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

- Sample Output

```
Case #1:
3
1
0
0
0
Case #2:
12
0
5
2
1
0
0
```

Problem.E　HE

Time Limit: 1000ms Memory Limit: 256MB

- Description

”Hi there, my name is HE.”

—

在某个秘密的计算机实验室中，超级人工智能计算机 HE 诞生了。

HE 可以在网络中搜索合适的程序代码进行自我升级。在某次升级的过程中，HE 得到了一堆软件代码，其中有很多注释和空行。注释是为了人类能够读懂代码而写的，而对于 HE 来说，它并不需要这些东西，因此它会首先删除掉代码中的**所有注释**以及**空行**。

这里定义了 2 种注释格式：

1. 行注释“//”：符号“//”之后直至行尾的所有内容都被认为是注释部分；

2. 段注释“/* ... */”：“/*”表示段注释开始，“*/”表示段注释结束，其间的所有内容都被认为是注释部分。

科学家发现 HE 的编译器分析部分出现了一些小问题，现在他们希望你能够编写程序帮助 HE 完成这一步骤。

- Input

输入文件只有一个，其中有许多包含注释的语言代码，请读至文件末尾（EOF）。

- Output

输出删除掉注释以及空行之后的代码。

注：

1. 若某一行删除完注释之后只剩下空格，则该行也将被视作一个空行，需要一起删除。

2. 输出的结果中**行末无空格**，若删除注释之后行末存在空格，则需要一起删除。

3. 保证题目数据都是可以通过正常编译的代码，样例数据中已经给出了大部分的情况，**详见样例**。

- Sample Input

```
#include<stdio.h>
//test
int main()
{
    printf("Helloworld.");
    return 0;
}
#include<stdio.h>
```

```
/* ----
---- */

/*/--/*/

int main()
{
    // test
    printf("Goodluck.");     // test
    return 0;   /* -- */
}
```

- Sample Output

```
#include<stdio.h>
int main()
{
    printf("Helloworld.");
    return 0;
}
#include<stdio.h>
int main()
{
    printf("Goodluck.");
    return 0;
}
```

Problem.F Go

Time Limit: 1000ms Memory Limit: 256MB

- Description

”Alpha Go is my little brother, I am much smarter than him.”

—

AlphaGo is a computer program developed by Google DeepMind in London to play the board game Go. In October 2015, it became the first Computer Go program to beat a professional human Go player without handicaps on a full-sized 19×19 board.

In March 2016, it beat Lee Sedol in a five-game match, the first time a computer Go program has beaten a 9-dan professional without handicaps. Although it lost to Lee Sedol in the fourth game, Lee resigned the final game, giving a final score of 4 games to 1 in favour of AlphaGo. In recognition of beating Lee Sedol, AlphaGo was awarded an honorary 9-dan by the Korea Baduk Association.

AlphaGo’s algorithm uses a Monte Carlo tree search to find its moves based on knowledge previously ”learned” by machine learning, specifically by an artificial neural network (a deep learning kind) by extensive training, both from human and computer play.

As of 2016, AlphaGo’s algorithm uses a combination of machine learning and tree search techniques, combined with extensive training, both from human and computer play. It uses Monte Carlo tree search, guided by a ”value network” and a ”policy network,” both implemented using deep neural network technology.

A limited amount of game-specific feature detection pre-processing (for example, to highlight whether a move matches a nakade pattern) is applied to the input before it is sent to the neural networks.

The system’s neural networks were initially bootstrapped from human gameplay expertise. AlphaGo was initially trained to mimic human play by attempting to match the moves of expert players from recorded historical games, using a database of around 30 million moves.

Once it had reached a certain degree of proficiency, it was trained further by being set to play large numbers of games against other instances of itself, using reinforcement learning to improve its play.

To avoid ”disrespectfully” wasting its opponent’s time, the program is specifically programmed to resign if its assessment of win probability falls beneath a certain threshold; for the March 2016 match against Lee, the resignation threshold was set to 20

AlphaGo’s March 2016 victory was a major milestone in artificial intelligence research. Go had previously been regarded as a hard problem in machine learning that was expected to be out of reach for the technology of the time. Most experts thought a Go program as

powerful as AlphaGo was at least five years away; some experts thought that it would take at least another decade before computers would beat Go champions.

Most observers at the beginning of the 2016 matches expected Lee to beat AlphaGo.

With games such as checkers (that has been "solved" by Chinook team), chess, and now Go won by computers, victories at popular board games can no longer serve as major milestones for artificial intelligence in the way that they used to. Deep Blue's Murray Campbell called AlphaGo's victory "the end of an era... board games are more or less done and it's time to move on."

When compared with Deep Blue or with Watson, AlphaGo's underlying algorithms are potentially more general-purpose, and may be evidence that the scientific community is making progress toward artificial general intelligence.

Some commentators believe AlphaGo's victory makes for a good opportunity for society to start discussing preparations for the possible future impact of machines with general purpose intelligence. (As noted by entrepreneur Guy Suter, AlphaGo itself only knows how to play Go, and doesn't possess general purpose intelligence: "It couldn't just wake up one morning and decide it wants to learn how to use firearms").

In March 2016, AI researcher Stuart Russell stated that "AI methods are progressing much faster than expected, (which) makes the question of the long-term outcome more urgent," adding that "in order to ensure that increasingly powerful AI systems remain completely under human control... there is a lot of work to do."

Some scholars, such as Stephen Hawking, warned (in May 2015 before matches) that some future self-improving AI could gain actual general intelligence, leading to an unexpected AI takeover; other scholars disagree: AI expert Jean-Gabriel Ganascia believes that "Things like 'common sense'... may never be reproducible", and says "I don't see why we would speak about fears. On the contrary, this raises hopes in many domains such as health and space exploration."

Computer scientist Richard Sutton "I don't think people should be scared... but I do think people should be paying attention."

—

Now, HE wants to play a Go game with you! Eee ... I'm sorry it is just a kidding.

Your task is much easier. Just count out which side has more pieces in the board.

- Input

The first line is an integer T, indicate there will be T test cases.

For each test cases:

There will be a 19*19 grid representing the Go board.

"#" means no piece, 0 means white piece and 1 means black piece.

- Output

For each test cases:

First output the case number like "Case #x:", x is the sequence number of that test case.

Then output "Black" if there are more black pieces in the board, or "White" otherwise.

If there are equal number of black pieces and white pieces, just output "Equal".

You don't have to care about whether some pieces will have to be removed from the board. Just count!

- Sample Input

```
1
####################
####################
###10###############
###00###############
####################
####################
####################
####################
####################
####################
####################
####################
####################
####################
##############111##
################0###
####################
####################
####################
```

- Sample Output

```
Case #1:
Equal
```

Problem.G Counting Star

Time Limit: 5000ms Memory Limit: 256MB

- Description

"Our Conquest is the Sea of Stars."

—

With the help of our powerful AI machine HE, scientists start an exciting project of sending space probes to the deep space.

Scientists have already got some pictures from Hubble Space Telescope, or I'd like to call them star maps. The star maps can be seen as a **(n+1)*(n+1) pixel grid**, from (0, 0) to (n, n), every grid point has a brightness.

In order to use some special analyzing algorithms, scientists decide to count **the total brightness of some right-angled triangle areas**. The triangle areas will always start from (0, 0).

Scientists will provide the star maps and tell you which areas should be counted and you are asked to answer it. The triangle areas will be given by its apexes position (0, 0), (x, 0), (x, y).

- Input

The first line is an integer T, indicates there will be T test cases.

For each test case:

The first line is an integer n ($n \leq 1000$), indicate the length and width of the star map.

Next is a n*n grid, pixels are numbered from (1, 1) to (n, n), **the lower left corner is (1, 1) and the upper right corner is (n, n)**. The rest parts of the (n+1)*(n+1) grid are "0" (Not included in the input file).

Then comes an integer m ($m \leq 100000$), indicate the number of queries.

Each of next m lines contains 2 integers (x, y). ($1 \leq x, y \leq n$)

- Output

For each test cases:

First output the case number like "Case #x:", x is the sequence number of that test case.

Then output m lines, each line is an integer number, means the brightness sum of the triangle areas by apexes position (0,0), (x,0), (x,y).

- Sample Input

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

- Sample Output

```
Case #1:
20
23
Case #2:
9
```

Problem.H Laser Cannon

Time Limit: 1000ms Memory Limit: 256MB

- Description

"To live or to die."

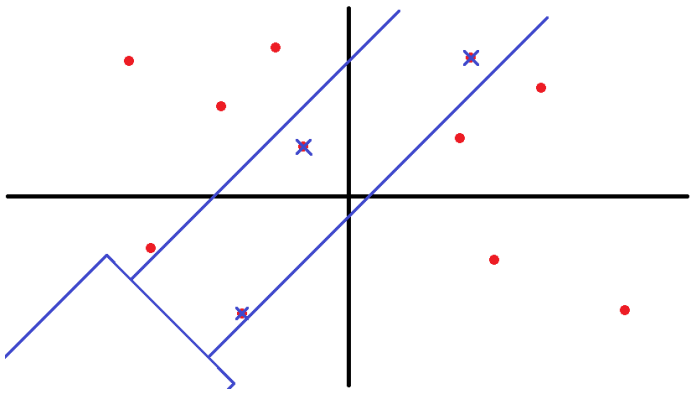
—

Space ship is going to launch, however scientists receive a bad news that there are many space trashes on the orbit.

For the space ship's safety, we have to destroy these space trashes first.

The target field can be seen as a **two dimensional surface**. Many trashes are located in the field and every trash has two properties: **value** and **mul**. The two properties are set for evaluating HE's behavior.

HE is given a laser cannon in order to destroy these trashes. The laser cannon is so powerful that **once an attacking angle is decided**, it is able to attack **a linear field with infinite length**. The linear field can be seen as two parallel lines with a tilt angle **a**, the distance between the two parallel lines can be **any value**. HE can choose anywhere to launch.



Every time HE uses laser cannon to launch an attack, the trashes **inside the linear field or even just on the boundary** will be destroyed. After that, HE will get a score depending on **those destroyed trashes**. The score HE will get is the sum of all those trashes' "value" times the average "mul" of all those trashes.

Just like this:

$$Score = \sum_{i=1}^{k} value[i] * \sum_{i=1}^{k} mul[i]/k$$

Now, HE can use the laser cannon for **several times**, and HE's task is to destroy **all the trashes in this field**. But notice that **the range of any two attacks should not be overlapped**. The final score is **the sum of each attack's score**, and of course, HE wants to get a maximum score in the end.

- Input

The first line is an integer T, indicate there will be T test cases.

For each test cases:

The first line is an integer n ($n \leq 2000$), indicate there are n trashes in the field.

Each of the next n lines contains 4 integer: x, y, value, mul, which describes the position of the trash[i], its "value" and its "mul". ($-10000 \leq x, y \leq 10000$,$1 \leq value, mul \leq 100$)

The next line is an integer a, means the tilt angle. ($0 \leq a \leq 180$)

- Output

For each test cases:

First output the case number like "Case #x:", x is the sequence number of that test case.

Then output a real number, the maximum score HE is able to get from this field. The answer should be accurate to 3 decimal places.

- Sample Input

```
1
3
1 3 3 1
2 1 2 2
3 4 2 1
45
```

- Sample Output

```
Case #1:
9.333
```

- Note

$\pi = 3.1415926$

Problem.I Mars City

Time Limit: 3000ms Memory Limit: 256MB

- Description

"Always at your service."

—

Supercomputers and AI machines are built to improve human's daily life.

This day, human have successfully landed on Mars. Scientists decide to build a City there for science research in the future.

They find that the area suitable for living is as much as 1000*1000 square meters, and will try to build N buildings there. Because of the unevenness of the Mars ground, the buildings may have to be built on the mountains.

Now, given all N buildings' three-dimensional position (x ,y ,z), z is the height of the building, scientists then want to build a supply net for the city.

The water and other resources are hard to get on the Mars. Scientists will choose **the highest building** as the supply center, space ships from the Earth will only send resource to the center and other buildings have to get supply from the center by **directed** supply lines.

Building a supply line costs a lot of money, so scientists only allow supply lines to be built **from higher buildings to the lower buildings or between two buildings that are of the same altitude**. If building A has already get supply from the supply center and building B's location is not higher than A, it is allowed to build a supply line from A to B. The resources will be transport from supply center to B through A.

The cost of building a supply line is the **Manhattan distance** of the two buildings times P dollar per meter. Now, HE has to solve this difficult problem to make the total cost minimized. Also, if there are many buildings of the highest altitude, HE has to choose one as the supply center to make the result minimized.

- Input

The first line is an integer T, indicate there will be T test cases.

For each test cases:

The first line is 2 integers n and P, indicates the number of buildings and the cost of money per meter. $(n, P \leq 1000)$

Each of the next n lines contains 3 integers x, y, z, meaning the three-dimensional position of the buildings. $(x, y, z \leq 1000)$

- Output

For each test cases:

First output the case number like "Case #x:", x is the sequence number of that test case.

Then output the minimum cost of the building project.

- Sample Input

```
2
3 10
1 1 10
1 2 5
2 1 5
4 10
1 1 10
1 3 10
3 2 5
4 2 6
```

- Sample Output

```
Case #1:
80
Case #2:
120
```

- Note

Manhattan distance between A(x1, y1, z1) and B(x2, y2, z2)

= abs(x1-x2)+abs(y1-y2)+abs(z1-z2)

Problem.J Magic Numbers

Time Limit: 1000ms Memory Limit: 256MB

- Description

网易在线笔试出了这么一道题：

Celin 一直认为万物皆数，他总会花上很多的时间去研究和数相关的一些问题。最近他在研究一种神奇的数，这种数包含以下 3 个特征：

1. 这个数至少包含 ('2', '3', '5') 中的任意一个数字;

2. 这个数不能出现'18'；

3. 这个数能被 7 整除。

如 217，280，1393，9520 均为同时满足三个条件的神奇的数。

而 140，798 不符合条件 (1)，518，1183 不符合条件 (2)，12，1727 不符合条件 (3)，这些数均不是 Celin 要找的神奇的数。

给出一个范围 [N, M]，Celin 想知道在范围内 (包括 N 和 M 两个边界在内) 一共有多少个符合条件的神奇的数。

出题人认为这个题目的数据范围太友好了，于是决定加大一点数据范围，卡掉部分不是特别好的做法。

- Input

每个输入数据包含多个测试点。第一行为测试点的个数 S <= 100，然后是 S 个测试点的数据。每个测试点为一行数据，该行包含两个数 N, M (1 <= N <= M <= 10210)，表示范围。

- Output

对应每个测试点的结果输出一行，表示范围内总共有多少个神奇的数, 由于答案非常大，你只需要输出答案 mod 1e9+7 即可。

- Sample Input

```
3
1 100
200 210
1000 1005
```

- Sample Output

```
6
2
0
```

Problem.K Kael's skill

Time Limit: 1000ms Memory Limit: 256MB

- Description

“出题老师目测出题 8000 分”——知名解说海涛。

由于万神 DOTA 打地实在太好了，所以他很快成为了网红，成为网红后的第一件事，就是把自己的卡尔练的对得起自己的名气。为了练习自己切换技能的手速，万神准备了 N 种不同的元素，这 N 种不同的元素分别被命名为 1，2，3,…N。现在万神要用这 N 种元素组成一个长度为 K 的序列，每个数字可以任意次的被取得，万神很好奇，这其中，元素 1 恰好出现 P 次的序列里面，字典序第 X 小的是多少？

例如，N=2，K=3，那么 1 恰好出现 1 次的序列有：

122，212，221。

第 1、2、3 小的分别是 122，212，221。

- Input

多组测试数据 (最多 20 组),以 EOF 结束。每组测试数据一行,共 3 个正整数 N(N<=9),K(K<=18),P(P<=K),X

- Output

每组测试数据输出一行，一个长度为 K 的序列表示答案，或者“poor god wan”表示无解。

- Sample Input

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

- Sample Output

```
122
212
221
```