Problem A. Mines

Description

Terrorists put some mines in a crowded square recently. The police evacuate all people in time before any mine explodes. Now the police want all the mines be ignited. The police will take many operations to do the job. In each operation, the police will ignite one mine. Every mine has its "power distance". When a mine explodes, any other mine within the power distance of the exploding mine will also explode. Please NOTE that the distance is Manhattan distance here.

More specifically, we put the mines in the Cartesian coordinate system. Each mine has position (x,y) and power distance d.

The police want you to write a program and calculate the result of each operation.

Input

There are several test cases.
In each test case:
Line 1: an integer N, indicating that there are N mines. All mines are numbered from 1 to N.
Line 2…N+1: There are 3 integers in Line i+1 (i starts from 1). They are the i-th mine's position (xi,yi) and its power distance di. There can be more than one mine in the same point.
Line N+2: an integer M, representing the number of operations.
Line N+3...N+M+2 : Each line represents an operation by an integer k meaning that in this operation, the k-th mine will be ignited. It is possible to ignite a mine which has already exploded, but it will have no effect.

1<=M<=N<=100000，0<=xi,yi<=10^9，0<=di<=10^9

Input ends with N=0.

Output

For each test case, you should print 'Case #X:' at first, which X is the case number starting from 1. Then you print M lines, each line has an integer representing the number of mines explode in the correspondent operation.

Sample Input

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

Sample Output

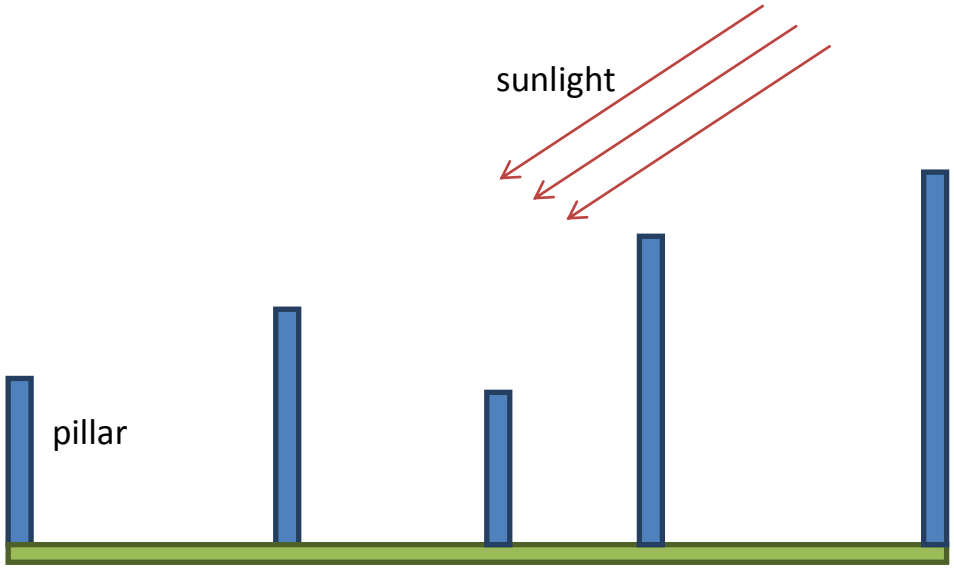
```
Case #1:
1
2
0
```

Problem B. Battery

Description

Recently hzz invented a new kind of solar battery. The battery is so amazing that the electric power generated by it can satisfy the entire village. People in the villager are all very happy since they can get free and green energy from now on. But the manager of a power company is sorrow about this. So he plans to take some action to obstruct the battery.

The battery can be regarded as a segment of ***L*** meters. And the manager plans to build ***n*** pillars on the battery.



The Battery

Like the picture above, the distance between pillar i and the battery's left end is ***Xi***, and its height is ***Hi.*** The thickness of all pillars can be ignored. When the sunlight is slant, some part of the battery will be sheltered by the pillars.

One meter battery exposed in the **vertical sunlight** for one hour will generate one unit of energy. If the sunlight is slant, the amount of energy generated should be multiplied by $\boldsymbol{sin\,\beta}$($\boldsymbol{\beta}$ is the angle between sunlight and horizontal line).

The sun rises from the infinite far left end of the horizon at 6 o'clock and goes down at the infinite far right end of the horizon at 18 o'clock. The sun is always infinite far away. So the sunlight is parallel, and $\boldsymbol{\beta}$ is $\frac{1}{2}\pi$ at 12 o'clock.

Please calculate the amount of energy generated by the battery between ***t1*** o'clock and ***t2*** o'clock

($6 \leq t1 < t2 \leq 18$).

Input

There are multiple test cases.
For each test case：
The first line contains two integer ***L***($10 \leq L \leq 100,000$) and***N***($4 \leq N \leq 1000$)***,*** indicating the length of the battery and the number of pillars.
The second line contains two integers, above mentioned ***t1*** and ***t2***($6 \leq t1 < t2 \leq 18$).
Then ***N*** lines follow, each containing two integers $Xi(0 \leq Xi \leq L)$ and $Hi(1 \leq Hi \leq 1000)$***,*** indicating the position and height of a pillar.

It is guaranteed that no two pillars will be in the same position. It is also guaranteed that there is a pillar on both end of the battery.
The input end with ***L=0, N=0.***

Output

For each test case, you should output a line with the energy described above. Output should be rounded to 5 digits after decimal point.

Sample Input

10 4
14 17
0 2
5 1
8 3
10 1
0 0

Sample Output

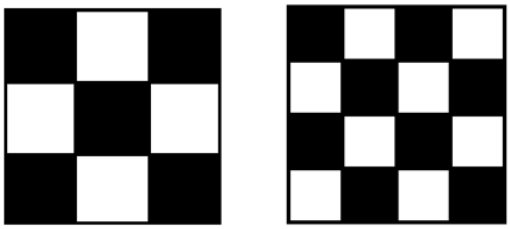
7.97188

Problem C. Magic Board

Description

Sths is a happy boy~

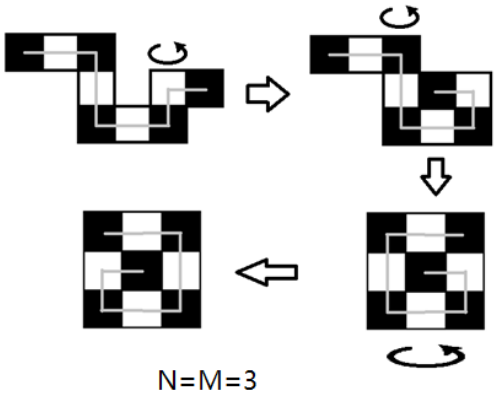
Sths has got a Magic Board for his birthday gift! A Magic board is an N*M sized grids which were painted by black and white. According to the parity of N and M, the Magic Board would be a little different, but it can be guaranteed that any two adjacent grids are painted by different colors, and the amount of black grid is not less than white ones.



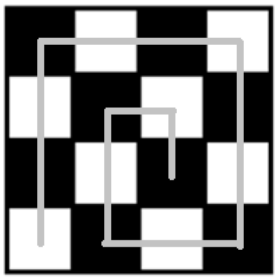
N=M=3 N=M=4

The Magic Board is called MAGIC because it is formed by a Chain. A Chain is a set of grids in which a grid has at most 2 adjacent grids and there are only 2 grids which has only 1 adjacent grid. Those two special grids are called the endpoints of the Chain.

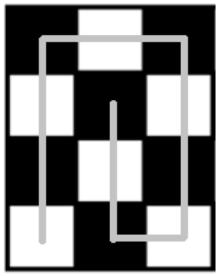
The joint between two adjacent grids is very flexible. It can be rotate by any angle. Here's an example of transforming a Chain into a Magic Board



N=M=3



N=M=4

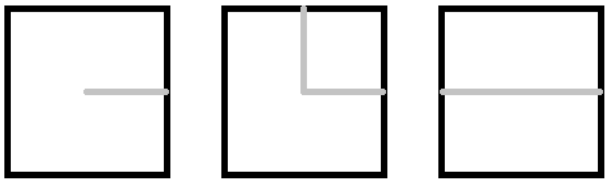


N=4 M=3

The Chain was connected by a Magic String. The existence of Magic String relies on the power of fengshui. But recently the fengshui in Beijing was ruined because there is a university installing air-conditionor (which also cause the HUGE RAIN in Beijing).So the Magic String disappears, and the Magic Board is totally fell apart.

Sths feels upset, because he really likes the Magic Board (since it can form a lot of things). So he is thinking about how to reconstruct it. The only thing Sths has got now is the separated grids. But surprisingly, Sths finds out that there are differences between these grids.

1. There are black grids and white grids.
2. There are three different grids in the same color because the Magic String goes through it in 3 different ways shown below:



Endpoints Adjacent Opposite

So there are 6 different kinds of grids. Now Sths has counted the amount of each kind of grids, he wants to know: by using the grids in his hand, how many kinds of legal Chains (which can form an N*M sized Magic Board) can be constructed.

We shall say two Chains is the same if and only if the standard expression of these two Chains is the same.

The standard expression is a set of numbers which decided by following method:

1. Starting from one of a Chain's endpoint.
2. Write down the color of the grids (1 for black and 0 for white) before direction changing.
3. Write down 2 then change direction and repeat Step 2 until reaching another endpoint of the Chain.
4. Choose the expression which lexicographical lower between the two expressions just generated since there are two endpoints.

For example, the standard expression of the example of “N=M=3” is “10120120120212”

(another expression is “10212012012012”, which is lexicographical greater than standard expression). And the standard expression of the example of “N=M=4” is “01012010210120120120212”.

Input

There are Multiple Test Cases
For each case, there will be six integer numbers in one line, N, M, BO, BA, WO, and WA, indicating the number of rows and columns, the amount of “Black and Opposite” grids, the amount of “Black and Adjacent” grids, the amount of “White and Opposite” grids, the amount of “White and Adjacent” grids.

2<=N*M<=30

The input end with a line of 0 0 0 0 0 0.

Output

For each case, output the kinds of legal Chains that can be constructed by given grids.

Sample Input

3 3 1 2 2 2
3 3 0 3 3 1
5 5 5 6 5 7
0 0 0 0 0 0

Sample Output

1
1
4

Problem D. A very hard Aoshu problem

Description

Aoshu is very popular among primary school students. It is mathematics, but much harder than ordinary mathematics for primary school students. Teacher Liu is an Aoshu teacher. He just comes out with a problem to test his students:

Given a serial of digits, you must put a '=' and none or some '+' between these digits and make an equation. Please find out how many equations you can get. For example, if the digits serial is "1212", you can get 2 equations, they are "12=12" and "1+2=1+2". Please note that the digits only include 1 to 9, and every '+' must have a digit on its left side and right side. For example, "+12=12", and "1++1=2" are illegal. Please note that "1+11=12" and "11+1=12" are different equations.

Input

There are several test cases. Each test case is a digit serial in a line. The length of a serial is at least 2 and no more than 15. The input ends with a line of "END".

Output

For each test case , output a integer in a line, indicating the number of equations you can get.

Sample Input

```
1212
12345666
1235
END
```

Sample output

```
2
2
0
```

Problem E. Worms

Description

Worms is a series of turn-based computer games. Players control a small platoon of earthworms across a deformable landscape, battling other computer- or player-controlled teams. The game feature bright and humorous cartoon-style animation and a varied arsenal of bizarre weapons.

During the course of the game, players take turns selecting one of their worms. They then use whatever tools and weapons available to attack and kill the opponents' worms. Over fifty weapons and tools may be available each time a game is played, and differing selections of weapons and tools can be saved into a "scheme" for easy selection in future games.

When most weapons are used, they cause explosions that deform the terrain, creating circular cavities. If a worm is hit by a weapon, the amount of damage dealt to the worm will be removed from the worm's initial amount of health. When a worm fall into the water or its health is reduced to zero, it dies.

In this problem, the terrain of a stone can be described as a simple polygon. The worms only use the time bombs. Once a time bomb is thrown, it is only attracted by the force of gravity. In other words, the flying track of the bomb is a parabola. When it reaches the stone (the polygon), the bomb does not blow off or stop immediately. It will still fly along the parabola regardless of the resistance of the stone due to its special character. The time bomb can only be triggered by the timer. When the preset time is used up, the bomb blows up and eliminates all the materials within its explosion range. You need to calculate the area of the eliminated materials of one explosion.

Input

There are multiple test cases.
The first line of a test case contains seven floating numbers $x0$, $y0$, $v0$, θ , t, g, R. ($x0$, $y0$) is the position of the worm who throws the bomb, which could be inside the polygon or outside the polygon (even in the sky or under the water). The initial value of velocity is v0 which forms aθ ($0 \leqslant \theta < 90$) angle with the positive direction of x-axis. The bomb is always thrown upwards. The preset time is t, which is also the time of flying. The value of acceleration of gravity of the worms' planet is g. The direction of gravity is the negative direction of y-axis. The explosion range is a circle and R is the radius.

The second line contains an integer n ($3 \leqslant n \leqslant 100$), which indicates the number of edges of the stone(simple polygon). The following n lines contain two real numbers xi and yi each, which describe the coordinates of a vertex. Two vertexes in adjacent lines are adjacent on the polygon.

The input contains multiple test cases. It is ended by "0 0 0 0 0 0 0".

Output

For each test case, output one line containing the area of the eliminated materials of the explosion rounded to two digits to the right of the decimal point.

Sample Input

```
0 0 15 45 10 2 10
3
100 0
200 0
100 100
0 0 0 0 0 0 0
```

Sample Output

```
228.74
```

Problem F. Aeroplane chess

Description

Hzz loves aeroplane chess very much. The chess map contains ***N+1*** grids labeled from 0 to N. Hzz starts at grid 0. For each step he throws a dice(a dice have six faces with equal probability to face up and the numbers on the faces are 1,2,3,4,5,6). When Hzz is at grid ***i*** and the dice number is ***x***, he will moves to grid ***i+x***. Hzz finishes the game when ***i+x*** is equal to or greater than ***N***.

There are also ***M*** flight lines on the chess map. The i-th flight line can help Hzz fly from grid ***Xi*** to ***Yi*** (**0**<***Xi***<***Yi***≤ ***N***)without throwing the dice. If there is another flight line from ***Yi***, Hzz can take the flight line continuously. It is granted that there is no two or more flight lines start from the same grid.

Please help Hzz calculate the expected dice throwing times to finish the game.

Input

There are multiple test cases.
Each test case contains several lines.
The first line contains two integers ***N(***$\mathbf{1} \leq N \leq \mathbf{100000}$) and $\mathbf{M}(\mathbf{0} \leq \mathbf{M} \leq \mathbf{1000})$***.***
Then M lines follow, each line contains two integers $\boldsymbol{Xi}, \boldsymbol{Yi}(\mathbf{1} \leq \boldsymbol{Xi} < Yi \leq N)$***.***
The input end with ***N=0, M=0.***

Output

For each test case in the input, you should output a line indicating the expected dice throwing times. Output should be rounded to 4 digits after decimal point.

Sample Input

```
2 0
8 3
2 4
4 5
7 8
0 0
```

Sample output

1.1667
2.3441

Problem G. GPA

Description

GPA(Grade-Point Average) is one way to measure students' academic performance in PKU. Each course has an integer credit, ranges from 1 to 99. For each course, you will get a score at the end of the semester, which is an integer ranges from 0 to 100. Then you can calculate the Grade-Point of this course with the following formula. (Your score is x and your Grade-Point is p, using real arithmetic)

$$p = 4 - \frac{3(100-x)^2}{1600}, 60 \le x \le 100$$
$$p = 0, \ \ 0 \le x \le 60$$

Then you can get the GPA with the following formula (the Grade-Point of course i is p_i, and the credit of course i is w_i).

$$GPA = \frac{\sum_{i=1}^{m} p_i \cdot w_i}{\sum_{i=1}^{m} w_i}$$

Now it is not far from the final exam, if you do not review, you can only get a basic score in each course.

You have n days to review. There are K classes in each day. For each class, only one course can be reviewed. After the review, your score in this course will exactly increase by 1. You can get more increment by spending more classes in this course. But the score may not exceed 100.

For some reasons, not any course can be reviewed in any class. Each day you can only review some of the courses.

Now you want your GPA to be as high as possible, and at the same time, you do not want to fail in any course. Please calculate the highest GPA you can get.

Input

The input consists of several test cases. Each test case begins with 3 integers N (0<=N<=40), K(0<K<=20), M (0<M<=20), representing the number of days, the number of classes in each day and the number of courses. Next line contains M integers representing credits of each course and M integers representing basic scores of each course (0<=score<=100). Next N lines contain an N*M matrix, the j^{th} element in i_{th} row means whether you can review course j in i^{th} day, 1 means you can review course j in i^{th} day, 0 means you cannot. The Input ends with 0 0 0.

Output

For each test case, output the highest possible GPA, round to 6 digits after decimal point. If you have to fail a course, output 0.000000 instead.

Sample Input

```
2 10 3
1 1 2
50 60 90
1 1 0
1 0 1
2 20 4
1 1 1 1
50 50 50 40
1 1 1 0
0 0 0 1
0 0 0
```

Sample Output

```
2.757813
0.000000
```

Problem H. Sum

Description

XXX is puzzled with the question below:

1, 2, 3, ..., n (1<=n<=400000) are placed in a line. There are m (1<=m<=1000) operations of two kinds.

Operation 1: among the x-th number to the y-th number (inclusive), get the sum of the numbers which are co-prime with p(1 <=p <= 400000).
Operation 2: change the x-th number to c(1 <=c <= 400000).

For each operation, XXX will spend a lot of time to treat it. So he wants to ask you to help him.

Input

There are several test cases.
For each case, the first line begins with two integers --- the above mentioned n and m.
Each the following m lines contains an operation.
Operation 1 is in this format: "1 x y p".
Operation 2 is in this format: "2 x c".

Output

For each operation 1, output a single integer in one line representing the result.

Sample Input

```
1
3 3
2 2 3
1 1 3 4
1 2 3 6
```

Sample Output

```
7
0
```

Problem I. Minimum Spanning Tree

Description

XXX is very interested in algorithm. After learning the Prim algorithm and Kruskal algorithm of minimum spanning tree, XXX finds that there might be multiple solutions. Given an undirected weighted graph with n (1<=n<=100) vertexes and m (0<=m<=1000) edges, he wants to know the number of minimum spanning trees in the graph.

Input

There are no more than 15 cases. The input ends by 0 0 0.
For each case, the first line begins with three integers --- the above mentioned n, m, and p. The meaning of p will be explained later. Each the following m lines contains three integers u, v, w (1<=w<=10), which describes that there is an edge weighted w between vertex u and vertex v(all vertex are numbered for 1 to n) . It is guaranteed that there are no multiple edges and no loops in the graph.

Output

For each test case, output a single integer in one line representing the number of different minimum spanning trees in the graph.
The answer may be quite large. You just need to calculate the remainder of the answer when divided by p (1<=p<=1000000000). p is above mentioned, appears in the first line of each test case.

Sample Input

```
5 10 12
2 5 3
2 4 2
3 1 3
3 4 2
1 2 3
5 4 3
5 1 3
4 1 1
5 3 3
3 2 3
0 0 0
```

Sample Output

4

Problem J. Family Name List

Description

Kong belongs to a huge family. Recently he got a family name list which lists all men (no women) in his family over many generations.

The list shows that the whole family has a common ancestor, let's call him Mr. X. Of course, everybody except Mr.X in the list is Mr. X's descendant. Everybody's father is shown in the list except that Mr. X's father is not recorded. We define that Mr. X's generation number is 0. His son's generation number is 1.His grandson's generation number is 2, and so on. In a word, everybody's generation number is 1 smaller than his son's generation number. Everybody's generation number is marked in some way in the list.

Now Kong is willing to pay a lot of money for a program which can re-arrange the list as he requires ,and answer his questions such as how many brothers does a certain man have, etc. Please write this program for him.

Input

There are no more than 15 test cases.
For each test case:
The first line is an integer N(1 <= N <= 30,000), indicating the number of names in the list.
The second line is the name of Mr. X.
In the next N-1 lines, there is a man's name in each line. And if the man's generation number is K, there are K dots('.') before his name.

Please note that :

1) A name consists of only letters or digits('0'-'9').
2) All names are unique.
3) Every line's length is no more than 60 characters.
4) In the list, a man M's father is the closest one above M whose generation number is 1 less than M.
5) For any 2 adjacent lines in the list, if the above line's generation number is G1 and the lower line' s generation number is G2, than G2 <= G1 +1 is guaranteed.

After the name list, a line containing an integer Q(1<=Q<=30,000) follows, meaning that there are Q queries or operations below.

In the Next Q lines, each line indicates a query or operation. It can be in the following 3 formats:

1) L
   Print the family list in the same format as the input, but in a sorted way. The sorted way means that: if A and B are brothers(cousins don’t count), and A's name is alphabetically smaller than B's name, then A must appear earlier than B.
2) b name
   Print out how many brothers does "name" have, including "name" himself.
3) c name1 name2
   Print out the closest common ancestor of "name1" and "name2". "Closest" means the generation number is the largest. Since Mr. X has no ancestor in the list, so it's guaranteed that there is no question asking about Mr. X's ancestor.

The input ends with N = 0.

Output

Already mentioned in the input.

Sample input

```
9
Kongs
.son1
..son1son2
..son1son1
...sonkson2son1
...son1son2son2
..son1son3
...son1son3son1
.son0
7
L
b son1son3son1
b son1son2
b sonkson2son1
b son1
c sonkson2son1 son1son2son2
c son1son3son1 son1son2
0
```

Sample output

```
Kongs
.son0
.son1
..son1son1
...son1son2son2
...sonkson2son1
..son1son2
..son1son3
...son1son3son1
1
3
2
2
son1son1
son1
```