Problem A. Ene's problem

Description

Ene was a girl lives in computer systems. When she was studying the periodic sequence $b_n = B^n \bmod P$ (B is an integer and P is a prime number, n >= 0), she chose a contiguous subsequence $b_s, b_{s+1}, \ldots, b_t$, and marked them as special numbers. Then she began to investigate another sequence $a_n = M \times A^n \bmod P$, (M and A are integers) and incidentally such a problem came to her mind: what is the minimum non-negative integer k such that a_k is a special number(k starts from 0) ?

As a result of Ene's ability to control a computer, she soon answered the question of herself. But as she had never taken any computer science courses, she could only enumerate all possible solutions to get the answer. Your task is to solve her questions faster, so your algorithm should possess a lot higher performance than hers.

Input

There are no more than 100 test cases in the input. In each case, you are given six integers M, A, B, s, t, P, satisfying $2 \leqslant A, B, M < P \leqslant 10^9$, $0 \leqslant s \leqslant t < P - 1$, and P is a prime number. The input ends by 0 0 0 0 0 0.

Output

For each test case, output your answer k in one line. If there's no solution, output "impossible" (no quotes).

Sample Input

2 3 5 1 6 11
2 3 8 1 6 11
0 0 0 0 0 0

Sample Output

impossible
1

Explanation of Sample

In the first test case, $b_1 = 5, b_2 = 3, b_3 = 4, b_4 = 9, b_5 = 1, b_6 = 5$, and $a_0 = 2, a_1 = 6, a_2 = 7, a_3 = 10, a_4 = 8, a_5 = 2$, it makes a cycle.

On the other hand, in the second test case, $b_1 = 8, b_2 = 9, b_3 = 6, b_4 = 4, b_5 = 10, b_6 = 3$, and $a_0 = 2, a_1 = 6 = b_3$. So that answer is 1.

Problem B. A Corrupt Mayor's Performance Art

Description

Corrupt governors always find ways to get dirty money. Paint something, then sell the worthless painting at a high price to someone who wants to bribe him/her on an auction, this seemed a safe way for mayor X to make money.

Because a lot of people praised mayor X's painting(of course, X was a mayor), mayor X believed more and more that he was a very talented painter. Soon mayor X was not satisfied with only making money. He wanted to be a famous painter. So he joined the local painting associates. Other painters had to elect him as the chairman of the associates. Then his painting sold at better price.

The local middle school from which mayor X graduated, wanted to beat mayor X's horse fart(In Chinese English, beating one's horse fart means flattering one hard). They built a wall, and invited mayor X to paint on it. Mayor X was very happy. But he really had no idea about what to paint because he could only paint very abstract paintings which nobody really understand. Mayor X's secretary suggested that he could make this thing not only a painting, but also a performance art work.

This was the secretary's idea:

The wall was divided into N segments and the width of each segment was one cun(cun is a Chinese length unit). All segments were numbered from 1 to N, from left to right. There were 30 kinds of colors mayor X could use to paint the wall. They named those colors as color 1, color 2 .... color 30. The wall's original color was color 2. Every time mayor X would paint some consecutive segments with a certain kind of color, and he did this for many times. Trying to make his performance art fancy, mayor X declared that at any moment, if someone asked how many kind of colors were there on any consecutive segments, he could give the number immediately without counting.

But mayor X didn't know how to give the right answer. Your friend, Mr. W was an secret officer of anti-corruption bureau, he helped mayor X on this problem and gained his trust. Do you know how Mr. Q did this?

Input

There are several test cases.

For each test case:

The first line contains two integers, N and M ,meaning that the wall is divided into N segments and there are M operations(0 < N <= 1,000,000 ; 0<M<=100,000)

Then M lines follow, each representing an operation. There are two kinds of operations, as described below:

1) P a b c

a, b and c are integers. This operation means that mayor X painted all segments from segment a to segment b with color c (0 < a<=b <= N, 0 < c <= 30).

2) Q a b

a and b are integers. This is a query operation. It means that someone asked that how many kinds of colors were there from segment a to segment b (0 < a<=b <= N).

Please note that the operations are given in time sequence.

The input ends with M = 0 and N = 0.

Output

For each query operation, print all kinds of color on the queried segments. For color 1, print 1, for color 2, print 2 ... etc. And this color sequence must be in ascending order.

Sample Input

```
5 10
P 1 2 3
P 2 3 4
Q 2 3
Q 1 3
P 3 5 4
P 1 2 7
Q 1 3
Q 3 4
P 5 5 8
Q 1 5
0 0
```

Sample Output

```
4
3 4
4 7
4
4 7 8
```

Problem C. Wang Xifeng's Little Plot

Description

《Dream of the Red Chamber》(also 《The Story of the Stone》) is one of the Four Great Classical Novels of Chinese literature, and it is commonly regarded as the best one. This novel was created in Qing Dynasty, by Cao Xueqin. But the last 40 chapters of the original version is missing, and that part of current version was written by Gao E. There is a heart breaking story saying that after Cao Xueqin died, Cao's wife burned the last 40 chapter manuscript for heating because she was desperately poor. This story was proved a rumor a couple of days ago because someone found several pages of the original last 40 chapters written by Cao.

In the novel, Wang Xifeng was in charge of Da Guan Yuan, where people of Jia family lived. It was mentioned in the newly recovered pages that Wang Xifeng used to arrange rooms for Jia Baoyu, Lin Daiyu, Xue Baochai and other teenagers. Because Jia Baoyu was the most important inheritor of Jia family, and Xue Baochai was beautiful and very capable , Wang Xifeng didn't want Jia Baoyu to marry Xue Baochai, in case that Xue Baochai might take her place. So, Wang Xifeng wanted Baoyu's room and Baochai's room to be located at two ends of a road, and this road should be as long as possible. But Baoyu was very bad at directions, and he demanded that there could be at most one turn along the road from his room to Baochai's room, and if there was a turn, that turn must be ninety degree. There is a map of Da Guan Yuan in the novel, and redists (In China English, one whose job is studying 《Dream of the Red Chamber》 is call a "redist") are always arguing about the location of Baoyu's room and Baochai's room. Now you can solve this big problem and then become a great redist.

Input

The map of Da Guan Yuan is represented by a matrix of characters '.' and '#'. A '.' stands for a part of road, and a '#' stands for other things which one cannot step onto. When standing on a '.', one can go to adjacent '.'s through 8 directions: north, north-west, west, south-west, south, south-east,east and north-east.

There are several test cases.

For each case, the first line is an integer N(0<N<=100) ,meaning the map is a N × N matrix.

Then the N × N matrix follows.

The input ends with N = 0.

Output

For each test case, print the maximum length of the road which Wang Xifeng could find to locate Baoyu and Baochai's rooms. A road's length is the number of '.'s it includes. It's guaranteed that for any test case, the maximum length is at least 2.

Sample Input

```
#.#
##.
..#
3
...
##.
..#
3
...
###
..#
3
...
##.
...
0
```

Sample Output

```
3
4
3
5
```

Problem D. Saving Tang Monk

Description

《Journey to the West》(also 《Monkey》) is one of the Four Great Classical Novels of Chinese literature. It was written by Wu Cheng'en during the Ming Dynasty. In this novel, Monkey King Sun Wukong, pig Zhu Bajie and Sha Wujing, escorted Tang Monk to India to get sacred Buddhism texts.

During the journey, Tang Monk was often captured by demons. Most of demons wanted to eat Tang Monk to achieve immortality, but some female demons just wanted to marry him because he was handsome. So, fighting demons and saving Monk Tang is the major job for Sun Wukong to do.

Once, Tang Monk was captured by the demon White Bones. White Bones lived in a palace and she cuffed Tang Monk in a room. Sun Wukong managed to get into the palace. But to rescue Tang Monk, Sun Wukong might need to get some keys and kill some snakes in his way.

The palace can be described as a matrix of characters. Each character stands for a room. In the matrix, 'K' represents the original position of Sun Wukong, 'T' represents the location of Tang Monk and 'S' stands for a room with a snake in it. Please note that there are only one 'K' and one 'T', and at most five snakes in the palace. And, '.' means a clear room as well '#' means a deadly room which Sun Wukong couldn't get in.

There may be some keys of different kinds scattered in the rooms, but there is at most one key in one room. There are at most 9 kinds of keys. A room with a key in it is represented by a digit(from '1' to '9'). For example, '1' means a room with a first kind key, '2' means a room with a second kind key, '3' means a room with a third kind key... etc. To save Tang Monk, Sun Wukong must get ALL kinds of keys(in other words, at least one key for each kind).

For each step, Sun Wukong could move to the adjacent rooms(except deadly rooms) in 4 directions(north, west, south and east), and each step took him one minute. If he entered a room in which a living snake stayed, he must kill the snake. Killing a snake also took one minute. If Sun Wukong entered a room where there is a key of kind N, Sun would get that key if and only if he had already got keys of kind 1,kind 2 ... and kind N-1. In other words, Sun Wukong must get a key of kind N before he could get a key of kind N+1 (N>=1). If Sun Wukong got all keys he needed and entered the room in which Tang Monk was cuffed, the rescue mission is completed. If Sun Wukong didn't get enough keys, he still could pass through Tang Monk's room. Since Sun Wukong was a impatient monkey, he wanted to save Tang Monk as quickly as possible. Please figure out the minimum time Sun Wukong needed to rescue Tang Monk.

Input

There are several test cases.

For each case, the first line includes two integers N and M(0 < N <= 100, 0<=M<=9), meaning that the palace is a N × N matrix and Sun Wukong needed M kinds of keys(kind 1, kind 2, ... kind M).

Then the N × N matrix follows.

The input ends with N = 0 and M = 0.

Output

For each test case, print the minimum time (in minutes) Sun Wukong needed to save Tang Monk. If it's impossible for Sun Wukong to complete the mission, print "impossible"(no quotes).

Sample Input

```
3 1
K.S
##1
1#T
3 1
K#T
.S#
1#.
3 2
K#T
.S.
21.
0 0
```

Sample Output

```
5
impossible
8
```

Problem E. Equality Test

Description

Numeric Regular Expression (NRE) is a simple version of regular expression. It is a useful tool for numeric string matching.

If S is an NRE, we often use function L(S) to represent the set of numeric strings matched by S. Note that L(S) maybe infinite.

NRE and function L can be recursively defined by the following rules:

1. Atom: A single digit (0,1,2,...,9) is an NRE, and L(d)={d}, where d=0,1,2,...9.
2. Nestification: If x is an NRE, then (x) is an NRE, and L((x))=L(x).
3. Concatenation: If x and y are NREs, then xy is an NRE, and L(xy)={ab | a∈x and b∈y}.
4. Option: If x and y are NREs, then x|y is an NRE, and L(x|y)= L(x)∪L(y).
5. Closure: If x is an NRE, then x* is an NRE, and L(x*)={ ε , L(x),L(xx),L(xxx),...}, here ε means "empty string". In other word, x* matches x zero or more times.

To avoid confusion, the order of operations (from high to low) is: nestification, closure, concatenation and option. Operations with higher order will be applied first, and same operations will be applied from left to right. For example, **01*** is equal to **0(1*)**, not **(01)***; **00|1** is equal to **(00)|1**, not **0(0|1)**.

Now give you two NERs r_1 and r_2, can you tell us whether $L(r_1)=L(r_2)$ or not?

Input

There are several test cases in the input.

The first line contains an integer T (1 <= T <= 50) -- the number of test cases.

Each test case contains two lines, the first line is r_1 and the second line is r_2

(1 <= $|r_1|$,$|r_2|$ <= 10). It is guaranteed that both r_1 and r_2 are correct NREs without any useless character.

Output

For each test case, output the answer "YES" or "NO" in a single line.

Sample Input

```
2
(1|2)*
(2|1)*
34|2
32|4
```

Sample Output

```
YES
NO
```

Problem F. Help!

Description

"Help! Help!"

While walking in the park, you suddenly hear someone shouting for help. You immediately realize that a person has fallen into the lake. As a brave man, you decide to save him.

You are really familiar with the terrain of the park. The park can be regarded as a 2D plane. And the lake is a convex polygon. At current, you are on (Xo, Yo), and the person is on (Xp, Yp). You also know that you can run *Vr* per second on the land, or swim *Vs* per second in the lake. Notice that you are allowed to run along the edge of the lake.

You are not good at swimming. You cannot stay in the lake longer than *Ts* second. And carrying another person will cut down your swimming speed by half.

Can you save the poor guy? What is the minimum time for you to reach him, and carry him back to the border of the lake?

Input

There are several test cases in the input.

The first line contains an integer T (1 <= T <= 20) -- the number of test cases.

For each case:

The first line contains three real numbers Ts, Vr, Vs. $0 < Ts < 10^8$, $0 < Vs < Vr < 10^8$.

The second line contains two real numbers Xo, Yo, indicate the position (Xo, Yo) of you at current.

The third line contains two real numbers Xp, Yp, indicate the position (Xp, Yp) of the person you are going to save.

The forth line contains only one integer N -- the number of vertices of the lake. 3 <= N <= 50000.

The follow N lines, each line contains two real numbers x, y, indicating one of the vertex (x, y) of the lake. The vertices of lake are listed in either clockwise or counter-clockwise order.

Each coordinate in the input does not exceed 10^6 by its absolute value. Your position is on the land and the person's is in the lake.

Output

For each test case, output the minimum time(in seconds) to save the poor person, rounded to two digits after the decimal point. If you cannot save he, output "-1" instead.

Sample Input

```
2
100 2 1
0 10
0 0
3
```

-1 1
1 1
0 -1
1 2 1
0 10
0 0
3
-1 1
1 1
0 -1

Sample Output

6.39
-1

Problem G. Defense Against Wolves

Description

Long long ago, there lived a lot of rabbits in the forest. Unfortunately, there were some wolves in the same forest. The rabbit kingdom was attacked by those wolves sometimes and they planned to defend against the wolves.

For each wolves' attack, the rabbit king would choose some rabbits(at least four rabbits) to make a defense team. We can regard each rabbit as a point in the three dimensional space. In the kingdom, there were no four rabbits stayed on a same plane.

During each attack, the number of wolves which would be killed was exactly equal to the number of rabbits which are ON the 3D convex hull of the defense team.

All defense teams should be different. But two different teams might share some rabbits. If the rabbit king couldn't choose a defense team any more, the rabbit kingdom would be destroyed.

Now the wolves want to calculate the number of wolves which would be killed if they wanted to destroy the rabbit kingdom.

Input

The input consists at most 50 test cases.

For each test case, the first line contains a integer n (4 <=n <= 50) denoting the number of rabbits.

The following n lines each contains three integers x, y, z (-1000 <= x, y, z <= 1000) denoting the coordinate of a rabbit. All rabbits' positions are different.

Output

For each test case, output one line containing the answer modulo 10^9+7.

Sample Input

```
4
0 0 0
1 0 0
0 1 0
0 0 1
5
0 0 0
10 0 0
0 10 0
0 0 10
3 3 3
```

Sample Output

```
4
24
```

Note

For the first case, the king could choose only one defense team.

For the second case, the available defense teams are {1,2,3,4,5}, {1,2,3,4}, {1,3,4,5}, {1,2,3,5}, {2,3,4,5}, {1,2,4,5}, and the number of rabbits which are on the 3d convex hull of those teams are all 4 and the answer is 6 * 4 = 24

Problem H. Relief grain

Description

The soil is cracking up because of the drought and the rabbit kingdom is facing a serious famine. The RRC(Rabbit Red Cross) organizes the distribution of relief grain in the disaster area.

We can regard the kingdom as a tree with n nodes and each node stands for a village. The distribution of the relief grain is divided into m phases. For each phases, the RRC will choose a path of the tree and distribute some relief grain of a certain type for every village located in the path.

There are many types of grains. The RRC wants to figure out which type of grain is distributed the most times in every village.

Input

The input consists of at most 25 test cases.

For each test case, the first line contains two integer n and m indicating the number of villages and the number of phases.

The following n-1 lines describe the tree. Each of the lines contains two integer x and y indicating that there is an edge between the x-th village and the y-th village.

The following m lines describe the phases. Each line contains three integer x, y and z indicating that there is a distribution in the path from x-th village to y-th village with grain of type z. (1 <= n <= 100000, 0 <= m <= 100000, 1 <= x <= n, 1 <= y <= n, 1 <= z <= 100000)

The input ends by n = 0 and m = 0.

Output

For each test case, output n integers. The i-th integer denotes the type that is distributed the most times in the i-th village. If there are multiple types which have the same times of distribution, output the minimal one. If there is no relief grain in a village, just output 0.

Sample Input

```
2 4
1 2
1 1 1
1 2 2
2 2 2
2 2 1
5 3
1 2
3 1
3 4
5 3
2 3 3
1 5 2
```

3 3 3
0 0

Sample Output

1
2
2
3
3
0
2

Note

For the first test case, the relief grain in the 1st village is {1, 2}, and the relief grain in the 2nd village is {1, 2, 2}.

Problem I. Rabbit's String

Description

Long long ago, there lived a lot of rabbits in the forest. One day, the king of the rabbit kingdom got a mysterious string and he wanted to study this string.

At first, he would divide this string into no more than k substrings. Then for each substring S, he looked at all substrings of S, and selected the one which has the largest dictionary order. Among those substrings selected in the second round, the king then choose one which has the largest dictionary order, and name it as a "magic string".

Now he wanted to figure out how to divide the string so that the dictionary order of that "magic string" is as small as possible.

Input

There are at most 36 test cases.

For each test case, the first line contains a integer k indicating the maximum number of substrings the king could divide, and the second line is the original mysterious string which consisted of only lower letters.

The length of the mysterious string is between 1 and 10^5 and k is between 1 and the length of the mysterious string, inclusive.

The input ends by k = 0.

Output

For each test case, output the magic string.

Sample Input

```
3
bbaa
2
ababa
0
```

Sample Ouput

```
b
ba
```

Note

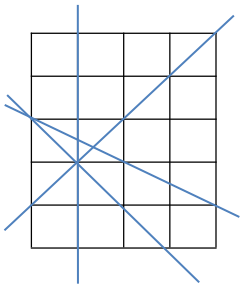
For the first test case, the king may divide the string into "b", "b" and "aa".

For the second test case, the king may divide the string into "aba" and "ba".

Problem J. Lines

Description

You play a game with your friend. He draws several lines on the paper with $n \times m$ square grids (see the left figure). After that, he writes down the number of lines passing through every integer coordinate in a matrix (see the right figure).



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

The number of lines passing though coordinate (i,j) is written in cell (i,j) in the right figure.(i,j both start from 0).

You are given the matrix written by your friend. You need to figure out the possible minimal number of lines your friend drew on the paper.

Input

The first line of the input contains an integer T indicating the number of test cases(0 < T <= 10).

For each test case, the first line contains two integers n, m ($1 \le n, m \le 50$) representing the size of the grids on the paper. The following (n+1)×(m+1) numbers is what your friend writes. It is guaranteed that the number of lines your friend draws does not exceed 14. Each line passes through integer coordinates at least three times.

Output

For each test case, you need to output the minimal number of lines your friend drew on the paper in a single line.

Sample Input

```
1
5 4
0 1 0 0 1
0 1 0 1 0
2 1 1 0 0
0 3 1 0 0
1 1 1 0 1
0 1 0 1 0
```

Sample Output

```
4
```