Problem A. Annual Beauty Contest

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

> "You are fair, I can't deny,
> But the Princess is the fairest
> And her beauty is the rarest!"
>
> A. S. Pushkin, "Tale of the Dead Princess and the Seven Knights"

It's time for the annual beauty contest! Imagine lines of awesome girls, showing their best looks... Unfortunately, the page margins are too narrow to fit their photos.

Being a judge of the contest is as hard as hell. The debates might take several days and a number of torn jacket buttons. To make things simpler, the Head of judges invented the following procedure.

First, every judge writes out a list of his preferences, i. e. some ordering of contestants. In this ordering, the first girl is the one the judge likes the least, the second is the least liked by him in absence of the first one, etc. After that, judges consequently place contestants: the first judge chooses the girl to be the last, the second one chooses the girl to be the pre-last and so on. After the n-th judge chooses someone, the first judge takes turn. The process continues until all girls have been assigned places. Each judge chooses the first (i. e., least liked) girl in his list that has not yet been chosen.

Vasya is the judge number k. He already knows the lists for every judge (telepathy, you know). And he does care of the only one contestant: Katya. He wants to arrange girls in his list in a way that Katya's final place would be as high as possible. Help him with this complicated task.

Input

Input consists of one or more test cases.

Each case starts with a single line containing integers n ($1 \le n \le 1\,000$), that denotes the number of judges, and k ($1 \le k \le 1\,000$), denoting the number of contestants. The second line contains two integers: Vasya's index among judges v ($1 \le v \le n$) and Katya's index among contestants x ($1 \le x \le k$). The following n lines contain judges' lists: k integers each. Each list starts with the girl that should be the last according to that judge. Vasya's list contains zeroes instead.

Input will be terminated with a test case with $n = k = 0$ which should not be processed.

The sums of n and k in all cases will not exceed $1\,000$ each.

Output

For each test case write the only integer: the highest rank Katya can achieve.

Adhere to the sample output format below as close as possible.

Example

standard input	standard output

`2 3`
`1 2`
`0 0 0`
`2 3 1`
`0 0`
	`Case #1: Katya's place can be 2.`

Problem B. Chess

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given is a toroidal three-dimensional chessboard $n \times n \times n$ $(2 \le n \le 4)$.

The cells have coordinates (x, y, z) $(0 \le x, y, z < n)$. Cell (x, y, z) is adjacent to cells $((x \pm 1) \bmod n,\ y,\ z)$, $(x,\ (y \pm 1) \bmod n,\ z)$ and $(x,\ y,\ (z \pm 1) \bmod n)$. Two distinct cells (x_1, y_1, z_1) and (x_2, y_2, z_2) are said to be *corner-neighbours* iff the maximum of "mod-distances" along x, y and z axes is 1. Here, the "mod-distance" between two integers a and b is the value $\min(|a - b|,\ n - |a - b|)$.

There are k $(0 \le k \le 4)$ aggressive black kings living on that chessboard, and they want to take the non-aggressive white king. Black and white move in turn. During a single move, player should move a single king of his colour to some corner-neighbouring cell. Exactly one king moves during a single move. Kings can take each other. This is done by moving to the cell where the other king is placed. One is only allowed to take king of the opposite colour. The king which is taken does not exist: it can't move or take other kings, and it doesn't occupy any cell. No two pieces can occupy the same cell.

You are given the initial coordinates of the pieces. The white king is the first to move. The question is: can the black kings manage to take him? If the answer is positive, you also have to find out the minimal number of moves to achieve that goal. The white king will resist this, so he'll try to get taken as late as possible.

Input

Input consists of one or more test cases.

Each test case starts with two integers n $(2 \le n \le 4$, the size of the chessboard) and k $(0 \le k \le 4$, the number of aggressive black kings). After that, $(k + 1) \cdot 3$ integers 0 through $n - 1$ follow: the coordinates of $k + 1$ kings (first 1 white king, then k black ones). Initially, all pieces occupy distinct cells.

Input will be terminated with a test case with $n = k = 0$ which should not be processed. There are no more than 10^4 tests.

Each integer in the input is followed by a nonempty sequence of spaces and/or newline characters.

Output

For each test case, write "`YES`" if the black kings manage to take the white king, and "`NO`" otherwise. In the first case, you should also write the minimal number of moves required if both players move optimally. You only have to count the black player's moves.

Adhere to the sample output format below as close as possible.

Example

standard input	standard output

2 4
0 0 0
0 0 1 0 1 0 1 0 0 1 1 1
4 4
2 2 2
0 0 0 0 0 1 0 1 0 0 1 1
0 0
	Case #1: YES 1
	Case #2: YES 4

Problem C. Concatenation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given the string S which consists of lowercase Latin letters. Consider string $T(S)$ which is the concatenation of all substrings of S in lexicographical order.

For example, if S = `aba`, the substrings are {`a`, `b`, `a`, `ab`, `ba`, `aba`}, the substrings in sorted order are {`a`, `a`, `ab`, `aba`, `b`, `ba`}, and thus $T(S)$ = `aaababababba`.

You have to find i-th character of the string $T(S)$.

Input

Input consists of one or more test cases.

Each case starts with a single line containing a positive integer m which denotes the number of queries. The next line contains string S ($1 \leq |S| \leq 5\,000$). The next line contains m integers a_i ($1 \leq a_i \leq |T(S)|$) denoting the queries themselves.

Input will be terminated with a test case with $m = 0$ which should not be processed.

The sum of m in all cases will not exceed $5\,000$.

The sum of lengths of all strings S will not exceed $5\,000$.

Output

For each test case, write a single line with m characters: answers to the queries. Adhere to the sample output format below as close as possible.

Example

standard input	standard output

`10`
`aba`
`1 2 3 4 5 6 7 8 9 10`
`1`
`x`
`1`
`0`
	`Case #1: aaababababba`
	`Case #2: x`

Problem D. Dictionary

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

> . . . Table of contents for a dictionary. . .
>
> The List of Most Useless Books

You've found a dictionary. There are several words listed lexicographically with a hieroglyphic translation. You've already scanned the dicionary. Unfortunately, the recognition software ignored all the hieroglyphs. Moreover, it ignored all the whitespace, leaving you with a long line of Latin letters.

As there's no linguistic interest left in the line, you can solve the following combinatory problem: count the number of ways to split the line into one or more distinct words such that the words will be listed lexicographically.

Input

Input consists of one or more test cases.

Each case consists of a single line of n lowercase Latin letters ($1 \le n \le 3\,000$).

Input will be terminated with a line containing a sinlge dash, which should not be processed.

The sum of n in all cases will not exceed $3\,000$.

Output

For each test case, write the number of ways to split the line. As this number might be huge, output it modulo $10^9 + 9$.

Adhere to the sample output format below as close as possible.

Example

standard input	standard output
`a`	`Case #1: There are 1 ways.`
`aa`	`Case #2: There are 1 ways.`
`ab`	`Case #3: There are 2 ways.`
`ba`	`Case #4: There are 1 ways.`
`abacaba`	`Case #5: There are 7 ways.`
`abracadabra`	`Case #6: There are 34 ways.`
`-`	

Problem E. Friends' Friends

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Social networks have recently changed the meaning of the word "friend". Once that meant a person who is open for you, who shares your interests, ideas, whatever. Now that's just an entry in a database, several bytes in your privacy settings.

But this is not the point. You are testing a new social network engine. To reproduce a certain bug, you need to find a pair of persons who don't have common friends.

Input

Input consists of one or more test cases.

Each case starts with a single line containing integer n ($1 \le n \le 1\,000$) that denotes the number of members of the social network.

The following n lines contain encoded graph of "friendship" relationships. Line number i describes i bits: j-th one is set to 1 iff people i and j are "friends" (note: this include the case when $i = j$, i.e. the person may be "friend" of self or not to be depending on value of this bit).

The graph is encoded using a kind of *Base64* scheme.See the following algorithm for details.

Bits are grouped per six. Each group of six bits is considered as a binary number 0 through 63. Numbers 0 through 25 are encoded with letters "A" through "Z", numbers 26 through 51 are encoded with letters "a" through "z", numbers 52 through 61 are encoded with digits "0" through "9", numbers 62 and 63 are encoded with characters "+" and "/", respectively. Each line containing number of bits that is not a multiple of six is padded with trailing zeroes. Each line is encoded independently. The first character on a line corresponds to the first six bits, the second one to the second group of six bits, and so on.

Input will be terminated with a test case where $n = 0$ which should not be processed.

The sum of n in all cases will not exceed $1\,000$.

Output

For each test case, write a pair of indices of members that have no common "friends", or an error message if there is no such pair. If there are multiple solutions, choose the one with the smallest possible first number. If there's still a tie, choose the smallest possible second number.

Adhere to the sample output format below as close as possible.

Example

standard input

```
3
A
g
Q
3
A
g
w
0
```

standard output

```
Case #1: Members 1 and 2 have no common friends.
Case #2: Social graph is too dense.
```

Problem F. Game Initialization

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

> "Three players take four tokens, and the fifth player keeps throwing out. Once the fox is eaten, it takes four moves back."
>
> "Foundling" movie

Well, it's time to set up the game field. There are n positions. Some of them are connected. Connections are made in a way that there's a single path from every position to every other one. There are also several tokens. Tokens may be placed onto the positions. There is only one restriction: two tokens can't be placed closer than c moves from each other.

To choose a fairly random placement, first find the number of ways to place the tokens. You may place any number of tokens, or even not place them at all.

And what about the goal of the game? I don't know. Maybe it's time for you to invent one.

Input

Input consists of one or more test cases.

Each test case starts with a single line containing integers n ($1 \le n \le 10\,000$) that denotes the number of positions, and c ($1 \le c \le \min(n, 500)$) that denotes the minimal distance between any pair of tokens.

The following $n - 1$ lines contain pairs of integers a_i, b_i denoting pairs of connected positions.

Input will be terminated with a test case with $n = c = 0$ which should not be processed.

The sum of n in all cases will not exceed $10\,000$.

Output

For each test case, write the number of ways to place the tokens modulo 10^6.

Adhere to the sample output format below as close as possible.

Example

standard input	standard output

5 2
1 2
1 3
1 4
4 5
2 2
1 2
2 1
1 2
0 0
	Case #1: 14
	Case #2: 3
	Case #3: 4

Problem G. Orders

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Vasya works for RIGHT (Research Institute of Government using Hash Tables). He is studying orders of some government of a country far far away.

In that country all towns are placed along some road. They are also numbered in the order of traversal. Initially, the quality level of life (QLoL) in every town equals zero.

After it, several orders have been issued. There is the only kind of orders: “the QLoL of all the towns i through j must become at least x”. Note that generally QLoL cannot be decreased.

There are also some official statements. They come in the following form: “average QLoL of towns i through j equals x”. Vasya wants you to help him check each of these statements in the following way: for each of them, you will be given only the pair (i, j), and your answer must contain the correct value of x.

You may assume that each order was executed, and at each moment of time, each town had the least possible QLoL satisfying all orders.

Input

The input consists of one or more test cases. Each test case starts with a line containing two integers n and k ”— the number of towns in the country and the number of orders. The following k lines will contain one event each. The event can be one of the following:

1. `^` i j x is an order: after it, all towns numbered i through j must have QLoL at least x ($1 \le x \le 10^9$, $1 \le i \le j \le n$).
2. `?` i j is an official statement; you should calculate the average QLoL for all towns numbered i through j ($1 \le i \le j \le n$).

The input will be terminated with a test case with $n = k = 0$ which should not be processed.

The sum of n’s in the whole input will not exceed 100 000. The sum of k’s in the whole input will not exceed 100 000.

Output

For each official statement, write a single line with average QLoL formatted as an irreducible fraction with smallest possible natural denominator. If the denominator of some value is 1, write it as an integer instead. See sample output for details.

Example

standard input	standard output
10 10 ? 1 10 ^ 1 10 1 ? 1 10 ^ 2 3 10 ^ 3 4 5 ? 2 2 ? 3 3 ? 4 4 ? 1 5 ? 1 10 0 0	0 1 10 10 5 27/5 16/5

Problem H. Oriental Puzzle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Vasya works for RIO (Research Institute of Oriental puzzles). He is studying a new puzzle consisting of several nodes and tubes partially filled with some liquid.

There are several nodes, plastic tubes connect some pairs of nodes. Each tube is made of a fabric that shines when the liquid passes through it. Two of the nodes are special: initially, first of them (source) contains all the liquid, and the other one (sink) is empty, but it should contain all the liquid after the puzzle will be solved (then one can swap their functions).

The goal is to arrange the tubes in such a way that after the liquid will be released from the source, all tubes will shine for some time (i. e. the liquid will be able to come through them), and finally, the liquid be collected in the sink.

Each tube should have some fixed orientation: one node will be higher, the other one will be lower. The liquid moves only from higher nodes to lower ones.

Help Vasya to solve that puzzle: find such orientation of tubes.

Input

The input consists of one or more test cases. Each test case consists of a single line with containing two integers n and k — the number of nodes and tubes ($2 \le n \le 10^5$, $0 \le k \le 10^5$). The following k lines contain descriptions of tubes: pair (a_i, b_i) means a tube connecting nodes a_i and b_i. The source will have number 1 and the sink will have number n. It is guaranteed that no tube connects a node with itself; nevertheless, there can be more than one tube between a pair of nodes.

The input will be terminated with a test case with $n = k = 0$ which should not be processed.

The total sums of n's and k's in all test cases will not exceed 100 000 each.

Output

For each test case, write a line with a word "Yes", if the orientation is possible and "No" otherwise. In the first case, write k more lines with descriptions of tubes in the same manner as in the input. The higher node in each description should go first.

Example

standard input	standard output
6 8	Yes
1 3	1 3
3 6	3 6
1 2	1 2
2 4	4 2
4 1	1 4
4 5	5 4
5 1	1 5
2 3	2 3
0 0	

Problem I. Data Packets

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 Mebibytes

Vasya works for RIDDLE (Research Institute of Data & Datagram probLEms). He is studying a new protocol of data exchange ”— an extension for RFC 1149 (IPoAC) and RFC 2549 (IPoAC with QoS). This extension has greater resistance for packet loss, however, another problem arises: two packets in a sequence might get mixed.

To simplify things, consider a pair of packets each containing a single integer ”— the number of the packet in the packet sequence. These packets were mixed during the transmission, and now instead of two numbers, there is a single sequence of digits. Vasya is sure that the difference between the numbers of packets is small, so he needs a program that can split the string into two numbers with the smallest possible difference.

Input

The input consists of one or more test cases. Each test case consists of a single line of $2 \le n \le 50$ digits 1 through 9.

The sum of lengths of all strings in the whole input doesn't exceed 100.

Output

For each test case write the only line containing two integers with the smallest possible difference which can be mixed, keeping internal order of digits, to obtain the input string. If there are multiple answers, output any of them.

Example

standard input	standard output

11
129
23917
	1 1
	12 9
	231 97

Problem J. Quartet Distance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

An *unrooted binary tree* is a tree with labeled leaves and unlabeled internal nodes with degree of all internal nodes equal to three. It can be proved that an unrooted binary tree with n leaves has exactly $2n-3$ edges.

Such trees are widely used in biology to display evolutionary relationships between species. Leaves represent living species which we can observe and internal nodes represent extinct ancestors of which we have no information.

One of the most common tasks is to find how similar are the two trees over the same set of species.

Consider four leaves, A, B, C, and D. They form a quartet $AB|CD$ in a tree T if there exists an edge that separates T into two subtrees T_1 and T_2 such that $A \in T_1$, $B \in T_1$, $C \in T_2$ and $D \in T_2$. Quartets are compared as partitions of a set of four vertices into two sets of two vertices in each. For example, quartets $AB|CD$, $BA|CD$, $DC|AB$ are all equal, whereas quartets $AB|CD$ and $AC|BD$ are different. Only leaves can form a quartet.

Let us denote the set of all quartets of a tree as $Q(T)$. Then we define the *quartet distance* between T_1 and T_2 as $d_{\mathrm{q}}(T_1, T_2) = |Q(T_1) \bigtriangleup Q(T_2)|$. Here, $A \bigtriangleup B$ is symmetric difference between sets A and B, that is, the set of elements which are either in A or in B, but not in both.

Given two trees, find the quartet distance between them.

Input

The first line of input contains an integer n, the number of leaves in each tree ($4 \le n \le 1000$). Then come $2n-3$ edges of the first tree, and after that, $2n-3$ edges of the second tree. Each edge occupies one line and contains a pair of numbers of vertices that it connects. Leaves are numbered by integers from 1 to n. Internal nodes are numbered by integers from $n+1$ to $2n-3$.

Output

On the first line, print the quartet distance $d_{\mathrm{q}}(T_1, T_2)$.

Example

standard input	standard output

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

Example explanation

In the example, $Q(T_1) = \{23|14,\ 23|15,\ 23|45,\ 45|12,\ 45|13\}$, $Q(T_2) = \{13|24,\ 13|25,\ 13|45,\ 45|12,\ 45|23\}$, $Q(T_1) \bigtriangleup Q(T_2) = \{23|14,\ 23|15,\ 13|24,\ 13|25\}$ and the distance is equal to 4.

Problem K. Rectangles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given a square matrix of integer numbers. Every cell of the matrix also has width and length, both equal to one.

You have to find such submatrix (subrectangle) that $\frac{S}{P}$ is maximal possible, where S is sum of all numbers in the submatrix and P is the length of the perimeter of the subrectangle corresponding to the submatrix.

Input

Input consists of one or more test cases.

Each case starts with a single line containing a single integer n ($1 \le n \le 300$) denoting the size of the matrix. The following n lines contain n integers each: the matrix itself. All elements of the matrix do not exceed 10^9 by absolute value.

Input will be terminated with a test case with $n = 0$ which should not be processed.

The sum of n in all cases will not exceed 300.

Output

For each test case, write the value of the function $\frac{S}{P}$ as precisely as possible. The value you output should differ from the exact answer by no more than 10^{-5}. After that, write x_1, y_1 and x_2, y_2: coordinates of the corners of the rectangle. The first corner should be the upper left corner, and the second one should be the lower right corner.

Adhere to the sample output format below as close as possible.

If there are several correct solutions, print any of them.

Example

standard input

```
2
100 100
1 1
0
```

standard output

```
Case #1: The maximal value is 33.333333333, rectangle corners are (1, 1) and (2, 1).
```

Problem L. Christmas Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

People of Manersk city are in preparation of celebrating the New Year. A big and beautiful Christmas Tree should be installed on the main square. The task of picking a tree was given to Vasya, a famous researcher from some research institute.

Forest warden Andrey knows the algorithm Vasya uses to pick a tree. Surely, he wants to find the highest spruce in the forest. However, the forest is large, and Vasya does not want to get lost in it. So before going to the forest, he will come to Andrey and ask which spruce should he pick. Then, Vasya will climb up that spruce and look around. From a spruce h meters high, he will see every tree in the square $2h$ meters $\times$ $2h$ meters centered at that spruce.

If Vasya sees a spruce higher than the one he just climbed, he will go to the highest of such spruces and climb up again. Vasya will continue doing so until he climbs up a spruce from which he can see no higher spruce. After that, he will climb down, cut that spruce down and carry it to the main square.

Forest warden Andrey knows everything about the forest and the spruces. In particular, he knows the coordinates and height of each of the n spruces in the forest. Help him save the forest, that is, show Vasya such spruce that Vasya will cut a spruce of height as small as possible.

Input

Input consists of one or more test cases. The first line of each test case contains n, the number of spruces in the forest. Next n lines contain three numbers each; i-th of these lines contains integers x_i, y_i and h_i — the coordinates and height of i-th spruce. All these integers do not exceed 10^9 by absolute value. There are no spruces of the same height in each particular test case.

Input is terminated by a line containing a single zero. The sum of n over all test cases does not exceed 50 000.

Output

For each test case, write two numbers on a single line: the number of spruce Andrey should show to Vasya and the height of the spruce which will be cut down as a result. In each test case, spruces are numbered from one to n in the order they are given in the input.

If there are several correct solutions, print any of them.

Example

standard input	standard output

`3`
`-1 1 2`
`-1 -1 3`
`0 3 4`
`0`
	`2 3`