UVa Online Judge

11830 Contract revision

For years, all contracts of the Association of Contracts for Modernization (ACM) were typed using an old typewriter machine.

Recently Mr. Miranda, one of the accountants of the ACM, realized that the machine had a failure in one, and only one, numerical digit. More specifically, the flawed digit, when typed, is not printed on the sheet, as if the corresponding key was not pressed. He realized that this could have changed the numerical representation of contract values. Worried about accounting, Mr. Miranda wants to know, from the original values agreed for the contracts (which he kept in handwritten notes) which values are actually represented in the contracts. For example, if the failed digit in the machine is 5, an agreed value of 1500 would be represented in the corresponding contract as 100, because the digit 5 would not be printed. Note that Mr. Miranda wants to know the numeric value represented in the contract, ie, in the same machine, the number 5000 corresponds to the numeric value 0, not 000 (as it actually appears in the contract).

Input

The input consists of several test cases, each in one line. Each line contains two integers D and N ($1 \leq D \leq 9$ and $1 \leq N < 10^{100}$), representing, respectively, the digit that has failed in the machine and the number that was originally agreed for the contract (which can be very large because of hiperinflation).

The last test case is followed by a line which contains only two zeros separated by white space.

Output

For each test case in the input your program must print one line containing a single integer, the numeric value represented in the contract.

Sample Input

```
5 5000000
3 123456
9 23454324543423
9 99999999991999999
7 777
0 0
```

Sample Output

```
0
12456
23454324543423
1
0
```

11831 Sticker Collector Robots

One of the favorite sports in RoboLand is the Robots Rally. This rally is practiced in a giant rectangular arena of square cells with dimensions N rows by M columns. Some cells are empty, some contain a sticker for the World Footbal Cup Album (much appreciated by artificial intelligences in RoboLand) and some are occupied by pillars which support the roof of the arena. During the rally the robots can occupy any cell in the arena, except those containing pillars, since they block the robot movement. The path of the robot in the arena during the rally is determined by a sequence of instructions. Each instruction is represented by one of the following characters: '`D`', '`E`' and '`F`', meaning, respectively, "turn 90 degrees to the right", "turn 90 degrees to the left" and "move forward one cell". Robots start the rally in some initial position in the arena and follow closely the sequence of instructions given (after all, they are robots!). Whenever a robot occupies a cell that contains a Cup sticker he collects it. Stickers are not replaced, that is, each scticker can be collected only once. When a robot tries to move into a cell which contains a pillar he stalls, remaining in the cell where he was, with the same orientation. The same happens when a robot tries to leave the arena.

Given the map of the arena, describing the position of pillars and sctickers, and the sequence of instructions of a robot, you must write a program to determine the number of stickers collected by the robot.

Input

The input contains several test cases. The first line of a test case contains three integers N, M and S ($1 \leq N, M \leq 100$, $1 \leq S \leq 5 \times 10^4$), separated by white spaces, indicating respectively the number of rows, the number of columns of the arena and the number of instructions to the robot. Each of the following N lines describes a cell line of the arena and contains a string with M characters. The first line to appear in the description of the arena is the one more to the North, the first column to appear in description of a cell line of the arena is the one more to the West.

Each cell in the arena is described by one of the following characters:

- '`.`' — normal cell;
- '`*`' — cell which contains a sticker;
- '`#`' — cell which contains a pillar;
- '`N`', '`S`', '`L`', '`O`' — cell where the robot starts the rally (only one in the arena). The letter represents the initial robot orientation (North, South, East and West, respectively).

The last line in the input contains a sequence of S characters among '`D`', '`E`' and '`F`', representing the instructions to the robot.

The last test case is followed by a line which contains only three numbers zero separated by a blank space.

Output

For each rally described in the input your program must print a single line containing a single integer indicating the number of stickers that the robot collected during the rally.

Sample Input

```
3 3 2
***
*N*
***
DE
4 4 5
...#
*#O.
*.*.
*.#.
FFEFF
10 10 20
....*.....
.......*..
.....*....
..*.#.....
...#N.*..*
...*......
..........
..........
..........
..........
FDFFFFFFEEFFFFFFEFDF
0 0 0
```

Sample Output

```
0
1
3
```

UVa Online Judge

11832 Account Book

The FCC (Foundation for Combating Corruption) dismantled a major corruption scheme in Nlogonia. During the operation, several account books, with notes documenting the illicit transactions carried out by the scheme, were seized by FCC agents.

Each page on the account books contains some transactions (income or expense, in nilogos, the local currency of Nlogônia, whose symbol is N$) and the cash flow resulting from transaco tions on that page. For example, if a page recorded the following transactions: an income of N$ 7, an income of N$ 2, an expense of N$ 3, an income of N$ 1 and an expense of N$ 11, the cash flow on that page would be 7 + 2 - 3 + 1 - 11 = -4.

However, to obstruct the work of the police, the offenders did not record in their account books the type of each transaction (income or expense). In the example above, the page would contain only the numbers 7, 2, 3, 1 and 11 (with no indication whether they were income or expense transactions). The cash flow for each page, however, was always recorded normally, with the signal (in this case, -4).

To guarantee that the offenders are convicted, prosecutors must be able to determine with certainty whether each transaction is an income or an expense. In the example above, transaction N$ 7 was certainly an income, and transaction N$ 11 was certainly an expense. But we cannot say anything about transactions N$ 2, N$ 3, and N$ 1. Transactions N$ 2 and N$ 1 could have been income and in this case transaction N$ 3 would have been an expense; or N$ 1 and N$ 2 could have been expenses and in this case transaction N$ 3 would be an income.

Many account books have a relatively large number of pages, with many transactions, making it is difficult for the police to process all the information. Therefore, they need a program that performs the task efficiently.

Input

The input consists of several test cases. The first line of a test case contains two integers N and F, indicating the number of transactions on the page ($2 \leq N \leq 40$) and cash flow for this page ($-16000 \leq F \leq 16000$). Each of the following N lines contains an integer T_i indicating the value of the i-th transaction ($1 \leq T_i \leq 1000$).

The last test case is followed by a line containing only two numbers zero separated by a space.

Output

For each test case the input your program must print a single line with N characters. The i-th character must be '+', if it is possible determine with certainty that the i-th transaction is an income, '-', if it is possible to determine with certainty that the i-th operation is an expense, and '?', if it is impossible to determine with certainty the type of transaction. If the cash flow recorded in the page cannot be obtained from the transactions recorded in the page, your program must print a single line containing the character '*'.

Sample Input

```
5 7
1
2
3
4
```

```
5
4 15
3
5
7
11
5 12
6
7
7
1
7
0 0
```

Sample Output

```
?+??+
*
+??-?
```

UVa Online Judge

11833 Route Change

The road system of a country connects all N cities so that it is possible to travel between any pair of cities using existing roads. Each road connects two different cities, is two-way and one has exactly one toll booth (a toll is paid for both directions of traffic). Roads intersect only in a city and no pair of cities is interconnected by two or more roads.

Dias Tranport offers a one-day parcel delivery service between cities. Each parcel must be transported from a city A to another city B. The management of Dias Transport defines, for each parcel, a *service route*, consisting of C cities and $C-1$ roads: the first city on the service route is the origin of the parcel, the final city is the destination of the parcel. The service route never passes twice through the same city, and the vehicle chosen to deliver a parcel can only travel by the service route defined.

One day, however, a vehicle broke down and was taken for repairs in a city that was not among the cities in its service route. The management of Dias Transport wants to know which is the lowest total cost, in terms of tolls, for delivering the parcel (that is, to take the vehicle from the city it was repaired to the destination city), but with an additional constraint: if at some point the vehicle reaches one of the cities that make up its service route, it should go back to following its service route.

Input

The input contains several test cases. The first line of a test case contais four integers N, M, C and K ($4 \leq N \leq 250$, $3 \leq M \leq N \times (N-1)/2$, $2 \leq C \leq N-1$ e $C \leq K \leq N-1$), representing, respectively, the number of cities, the number of roads, the number of cities in the service route and the city where the vehicle was taken for repair. The cities are identified by integers from 0 to $N-1$. The service route is $0, 1, \ldots, C-1$, that is, the origin is 0, from 0 goes to 1, from 1 to 2 and so on, until the destination $C-1$. The next M lines describe the road system. Each of those lines describes one road and contains three integers U, V and P ($0 \leq U$, $V \leq N-1$, $U \neq V$, $0 \leq P \leq 250$), indicating that there exists a road connecting cities U and V with a toll of cost P.

The last test case is followed by a line containing four zeros separated by blank spaces.

Output

For each test case, your program should print a single line, containing a single integer, the minimum total toll cost for the vehicle to reach the destination city.

Sample Input

```
4 6 3 3
0 1 10
1 2 10
0 2 1
3 0 1
3 1 10
3 2 10
6 7 2 5
5 2 1
2 1 10
1 0 1
3 0 2
```

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

Sample Output

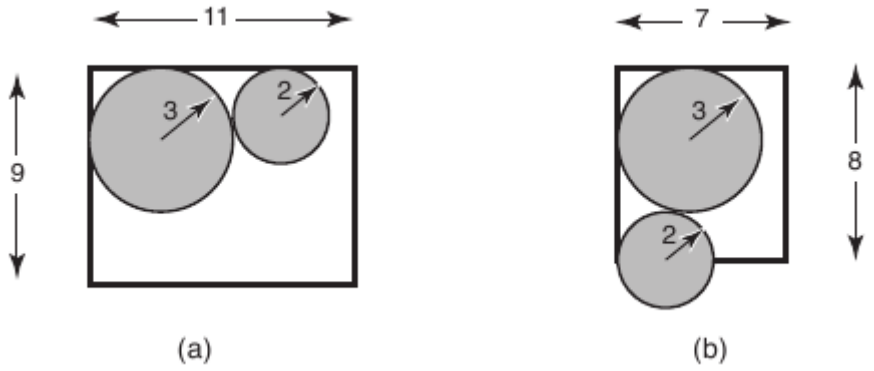
```
10
6
6
```

UVa Online Judge

11834 Elevator

The FCC (Factory of Cylinders of Carbon) manufactures various types of cylinders of carbon. FCC is installed on the tenth floor of a building, and uses the several building's elevators to transport the cylinders. For security, the cylinders must be transported in the upright position, and since they are heavy, at most two cylinders can be transported in a single elevator ride. The elevators have the shape of a parallelepiped and their height is always greater than the height of the cylinders.

To minimize the number of elevator trips to transport the cylinders, the FCC wants, whenever possible, to put two cylinders in the elevator. The figure below illustrates, schematically (top view) a case where this is possible (a), and a case where this is not possible (b):



As there is a very large amount of elevators and types of cylinders, FCC hired you to write a program that, given the dimensions of the elevator and of the two cylinders, determines whether it is possible to put the two cylinders in the elevator.

Input

The input contains several test cases. The first and only line of each test case contains four integers L, C, R_1 and R_2 , separated by blanks, indicating the width ($1 \leq L \leq 100$) and the length ($1 \leq C \leq 100$) of the elevator and the radii of the cylinders ($1 \leq R_1, R_2 \leq 100$).

The last test case is followed by a line containing four zeros separated by blanks.

Output

For each test case your program should print a single line with a single character, '`S`' if you can put the two cylinders in the elevator and '`N`' otherwise.

Sample Input

```
11 9 2 3
7 8 3 2
10 15 3 7
8 9 3 2
0 0 0 0
```

Sample Output

```
S
N
N
S
```

UVa Online Judge

11835 Formula 1

The Formula 1 season consists of a series of races, known as Grand Prix, organized by the International Federation of Automobile (FIA). The results of each Grand Prix are combined to determine Pilots' World Championship. More specifically, for each race some points are distributed to pilots, depending on their classification in the race. At the end of the season, the pilot who has earned the most points is declared the World Champion.

Formula 1 organisers change constantly the competition rules, aiming to provide more excitement to fans. One rule modified for the 2010 season was the distribution of points in each Grand Prix. Since 2003, the scoring rule rewarded the top eight pilots, according to the following table

Place	1	2	3	4	5	6	7	8
Points	10	8	6	5	4	3	2	1

That is, the winning driver received 10 points, second place received 8 points, and so on. In the 2010 season the top ten will receive points, obeying the following table:

Place	1	2	3	4	5	6	7	8	9	10
Points	25	18	15	12	10	8	6	4	2	1

The change in the scoring system led to much speculation about what would have been the effect to the World Championship in the past if the new score had been used. For example, would Lewis Hamilton have been champion in 2008, considering he and Felipe Massa were separated by just one point? To end the speculation, FIA hired you to write a program that, given the results of each race of a season determines the World Champion for different scoring systems.

Input

The input contains several test cases. The first line of a test case contains two integers G and P separated by a blank space, indicating the number of Grand Prix ($1 \leq G \leq 100$) and the number of pilots ($1 \leq Q \leq 100$). Pilots are identified by integers from 1 to P. Each of the following G lines indicates the result of a race, and contains Q integers separated by spaces. On each line, the (i)-th number indicates the order of arrival of pilot i in the race (the first number indicates the order of arrival of a pilot 1 in that race, the second number indicates the order of arrival of pilot 2 in that race and so on). The next line contains a single integer S indicating the number of scoring systems ($1 \leq S \leq 10$). After that, each of the following lines S contains a description of a scoring system. The description of a scoring system begins with an integer K ($1 \leq K \leq P$), indicating the last finishing order to receive points, followed by a blank space, followed by K integers $k_0, k_1, \ldots, k_{n-1}$ ($1 \leq k_i \leq 100$) separated by spaces, indicating the number of points to be assigned (the first integer indicates the points for first place, the second integer indicates the points for second place and so on).

The last test case is followed by a line containing only two zeros separated by a blank space.

Output

For each scoring system in the input your program must print one line, containing the identifier of the World Champion. If more than one pilot are World Champions (ie, if there is a tie), the line must contain all World Champions, in increasing order of identifier, separated by a space.

Sample Input

```
1 3
3 2 1
3
3 5 3 2
3 5 3 1
3 1 1 1
3 10
1 2 3 4 5 6 7 8 9 10
10 1 2 3 4 5 6 7 8 9
9 10 1 2 3 4 5 6 7 8
2
5 5 4 3 2 1
3 10 5 1
2 4
1 3 4 2
4 1 3 2
2
3 3 2 1
3 5 4 2
0 0
```

Sample Output

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

UVa Online Judge

11836 Star War

Long ago in a galaxy far, far away, there was an empire that dominated all others. A rebel alliance, unhappy with this situation, decided to fight these forces, with the objective of restoring democracy and peace for all nations.

Captain Cael, one of the rebel commanders, is sailing through space with his space cruiser, when he suddenly detects the presence of a ship of the Empire (according to the aesthetic standards of the time, all ships are tetrahedra). After a moment of surprise, Cael realizes he is in firing range and may place a cannon at any point of his ship.

As the power of his gun is fixed, Cael wants to position the cannon so that the distance traveled by the energy beam to the Empire's ship is minimal, to maximize the damage. Therefore, he asked that you, sub-master Cin Talig, compute the shortest distance between the rebel spacecraft and the spacecraft of the Empire.

Input

The input contains several test cases. The first line of the input contains an integer T, indicating the number of test cases. Each of the T test cases consists of eight lines, each line describing the coordinate of a vertex of a ship. The first four lines describe the vertices of the rebel spacecraft, the following four lines describe the vertices of the spacecraft of the Empire.

Each coordinate description is a line containing three integers X, Y, Z indicating the coordinate of the vertex in space ($-10^3 \leq X \leq 10^3$, $-10^3 \leq Y \leq 10^3$, $-10^3 \leq Z \leq 10^3$). The four corners of each ship always define a tetrahedron of nonzero volume, and the two ships are always disjoint.

Output

For each test case in the input your program must print a line containing a single number, printed with precision of two decimal digits, indicating the minimum distance between the two spacecrafts. The distance between the two ships is always greater than zero.

Sample Input

```
3
2 -1 -1
0 -1 -3
1 1 -4
1 1 -2
0 5 -1
2 5 1
1 3 2
1 3 0
1 0 -6
-5 0 -4
-2 6 -5
-2 2 -2
1 0 3
-5 0 5
-2 2 7
```

```
-2 -4 7
4 -4 -2
-2 -4 -4
1 4 -3
1 0 -4
-2 4 -1
4 4 1
1 -4 0
1 0 -1
```

Sample Output

```
2.83
6.03
1.90
```

UVa Online Judge

11837 Musical Plagiarism

The musical notes are the basic units of Western music. Each note is associated with a frequency. Two notes for which the fundamental frequencies have a ratio of power of 2 (one is half of the other, one is double of the other, etc..) are perceived as very similar. Therefore, any notes with that kind of relationship are given the same name, as described below.

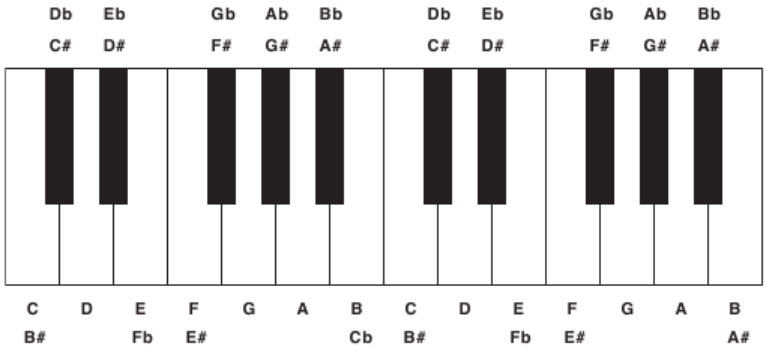
There are twelve basic notes in a sequence of increasing frequency, each note separated from the previous note by the same distance in the musical scale (this distance is called a semitone). Seven of these twelve notes are represented by letters of the alphabet (A, B, C, D, E, F and G). The table below shows the distance, in semitones, between the notes.

Notes	A-B	B-C	C-D	D-E	E-F	F-G	G-A
Number of semitone	2	1	2	2	1	2	2

Notice that there are five notes that are not represented by letters of the alphabet: the notes between A and B, between C and D, between D and E, betwen F and G and between G and A.

Notes can be modified by two accidentals, called sharp and flat, represented respectively by the symbols '`#`' and '`b`'. A sharp raises the note a semitone, a flat lowers the note a semitone. A note with an accidental is denoted by the name of the note followed by the accidental symbol. Notice that with this scheme we can represent all twelve notes.

The figure below illustrates the name of the notes in accordance with the scheme described above, in a fragment of a piano keyboard.



A melody can be represented by a sequence of notes. For example,

```
A A D C# C# D E E E F# A D G# A
```

is a well known melody. Note however that as the distances between the semitones are always equal, the same melody can be written starting with another note (we say that the melody is in another key):

```
B B E D# D# E Gb Gb Gb G# B E A# B
```

Your neighbor is a famous composer who suspects someone has plagiarized one of her songs. She asked your help to write a program that, given the sequence of notes of the melody in her song, and the sequence of notes of a suspicious snippet of melody, determines if the suspicious snippet occurs, in some key, in her song.

Input

The input consists of several test cases. The first line of a test case contains two integers M and T ($1 \leq M \leq 10^5$, $1 \leq T \leq 10^4$, $T \leq M$), indicating the number of notes in the song suspected of having been plagiarized and in the suspect snippet. Each of the next two lines contains M and T notes, respectively, indicating the notes of the song and of the suspect snippet.

Notes in each line are separated by one space; each note is one among '`A`', '`B`', '`C`', '`D`', '`E`', '`F`' or '`G`', possibly followed by an accidental sign: '`#`' for sharp or '`b`' for flat.

The last test case is followed by a line containing only two zeros separated by a blank space.

Output

For each test case your program must print a single line containing one character, '`S`' in case the song has been plagiarized by the text, or '`N`' in case the song has not been plagiarized by the text.

Sample Input

```
16 4
D G A B C D G G G C D E F# G C C
G G C D
12 2
C C# D D# E F F# G G# A A# B
C D
12 2
C Db D Eb E F Gb G Ab A Bb B
C D
4 3
C E G Bb
D F# A
0 0
```

Sample Output

```
S
N
N
S
```

UVa Online Judge

11838 Come and Go

In a certain city there are N intersections connected by one-way and two-way streets. It is a modern city, and several of the streets have tunnels or overpasses. Evidently it must be possible to travel between any two intersections. More precisely given two intersections V and W it must be possible to travel from V to W and from W to V.

Your task is to write a program that reads a description of the city street system and determines whether the requirement of connectedness is satisfied or not.

Input

The input contains several test cases. The first line of a test case contains two integers N and M, separated by a space, indicating the number of intersections ($2 \leq N \leq 2000$) and number of streets ($2 \leq M \leq N(N-1)/2$). The next M lines describe the city street system, with each line describing one street. A street description consists of three integers V, W and P, separated by a blank space, where V and W are distinct identifiers for intersections ($1 \leq V, W \leq N, V \neq W$) and P can be 1 or 2; if $P = 1$ the street is one-way, and traffic goes from V to W; if $P = 2$ then the street is two-way and links V and W. A pair of intersections is connected by at most one street.

The last test case is followed by a line that contains only two zero numbers separated by a blank space.

Output

For each test case your program should print a single line containing an integer G, where G is equal to one if the condition of connectedness is satisfied, and G is zero otherwise.

Sample Input

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

Sample Output

```
1
1
```

```
0
0
```

UVa Online Judge

11839 Optical Reader

Professor John decided to apply only multiple-choice tests to his students. In each test, each question will have five alternatives (A, B, C, D and E), and the teacher will distribute one answer sheet for each student. At the end of the test, the answer sheets will be scanned and processed digitally to obtain the test score for each student. Initially, he asked a nephew, who knows computer programming, to write a program to extract the alternatives marked by the students in the answer sheets. The nephew wrote a good piece of software, but cannot finish it because he needs to study for the ICPC contest.

During processing, the answer sheets are scanned in gray levels between 0 (full black) and 255 (total white). After detecting the position for the five rectangles corresponding to each of the alternatives of a question, the software calculates the average pixel gray level for each rectangle, returning an integer value corresponding to each alternative. If the rectangle was filled properly the average value is zero (all black). If the rectangle was left blank the average value is 255 (white total). Thus, ideally, if the average values for each alternative of a question are (255, 0, 255, 255, 255), we know that the student has marked alternative B for that question. However, as answer sheets are processed individually, the average gray level for a completely filled rectangle is not necessarily 0 (may be higher), and the value for a rectangle not filled is not necessarily 255 (may be less). Professor John determined that rectangle average gray levels would be divided into two classes: those with values equal or lower to 127 are considered black and those with values higher than 127 will be considered white.

Obviously, not necessarily all questions of all sheets are marked correctly. It can happen that a student makes a mistake and marks more than one alternative for the same question, or does not mark any alternative. In such cases, the answer to the question should be disregarded.

Professor John now needs a volunteer to write one program that, given the average gray level values of the five rectangles corresponding to the alternatives of a question, determines which alternative was marked, or whether the answer to the question should be disregarded.

Input

The input contains several test cases. The first line of a test case contains an integer N indicating the number of questions in the answer sheet ($1 \leq N \leq 255$). Each of the N following lines describes the response to a question and contains five integers A, B, C, D and E, indicating the values of average gray levels for each alternative ($0 \leq A, B, C, D, E \leq 255$).

The last test case is followed by a line containing only a number zero.

Output

For each test case the input your program must print N lines, each line corresponding to a question. If the answer to the question was correctly filled in the answer sheet, the line should contain the alternative selected (‘`A`’, ‘`B`’, ‘`C`’, ‘`D`’ or ‘`E`’). Otherwise, the line should contain the character ‘`*`’ (asterisk).

Sample Input

```
3
0 255 255 255 255
255 255 255 255 0
255 255 127 255 255
4
200 200 200 0 200
```

```
200 1 200 200 1
1 2 3 4 5
255 5 200 130 205
0
```

Sample Output

```
A
E
C
D
*
*
B
```

UVa Online Judge

11840 Tic-tac-toe

Tic-tac-toe is one of the oldest games of mankind. The first records of this game are from the first century BC, during the Roman Empire. John and Mary love to play the game, but after a while they decided to play a variant of the old traditional game, *Tic-tac-toe 1-D*.

Tic-tac-toe 1-D is a game played by two players on a board $1 \times N$; initially, all the squares of the board are empty. Players take turns drawing a cross on an empty square. The first player to complete a sequence of three or more crosses in contiguous squares wins the game.

Mary soon realized that, depending on the game situation, being her turn, she can guarantee she will win, regardless of John's moves. This is relatively easy for smaller boards, but for larger boards, after several moves, this task is more difficult. So, she asked you to write a program that, given the state the board, decides whether there exists a winning strategy.

Input

The input contains several test cases. The first line of a test case contains an integer N, indicating the size of the board ($3 \leq N \leq 10^4$). The next line contains a sequence of N characters indicating which squares of the board have been marked: a '`.`' indicates that the corresponding square is empty, while a '`X`' indicates that the square already has a cross drawn. The input never contains three contiguous '`X`'.

The last test case is followed by a line containing a single number zero.

Output

For each test case in the input your program must print a single line containing a single character, '`S`' if Mary has a winning strategy or '`N`' otherwise.

Sample Input

```
5
.....
5
..X..
6
X.X.X.
12
............
0
```

Sample Output

```
S
N
S
N
```