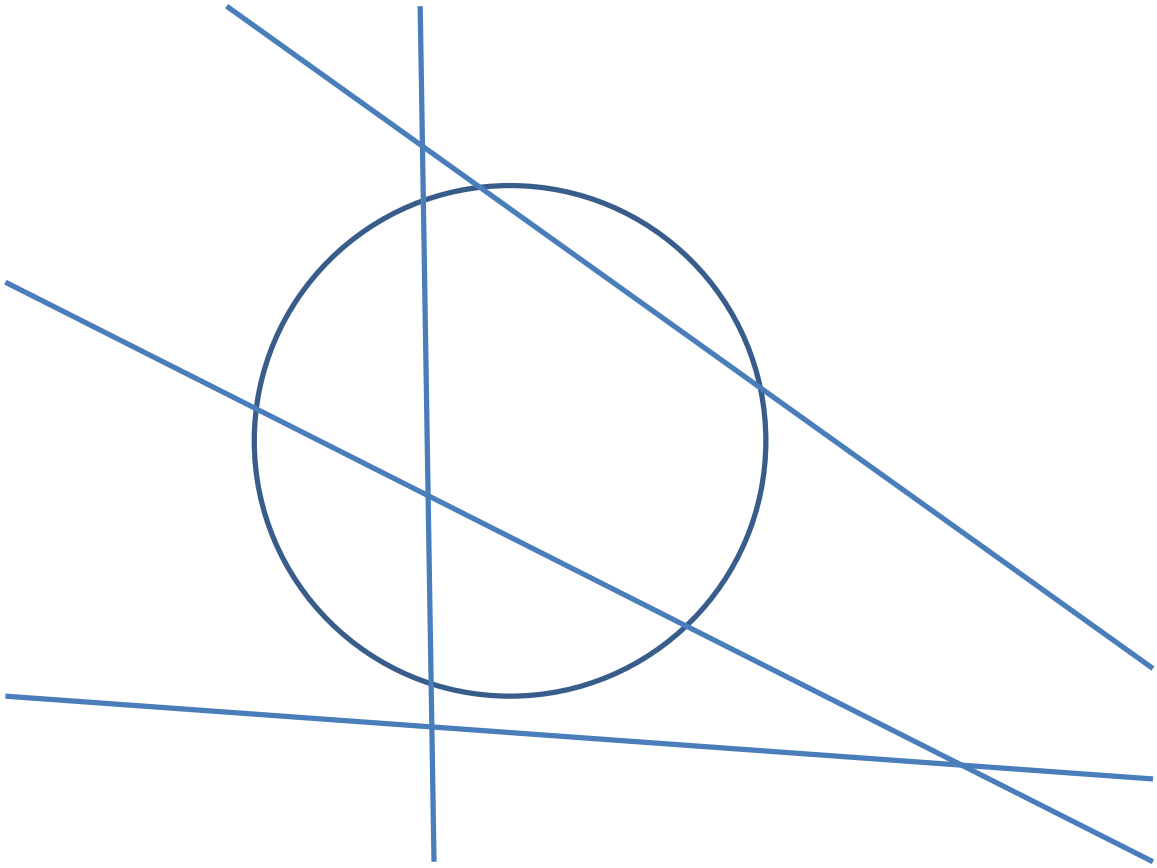
A: Cut the Cake

Given a circle, and a list of lines, how many parts has the circle been cut into? In the following example, four lines cut the circle into five parts.



Input

There will be several test cases in the input. Each test case will begin with four integers, ***r*** (1≤**r**≤1,000), ***x***, ***y*** (-1,000≤***x***,***y***≤1,000), and ***n*** (0≤***n***≤1,000), where ***r*** is the radius of a circle, ***x*** and ***y*** are the coordinates of the center of the circle, and ***n*** is the number of lines. On each of the next ***n*** lines will be four integers, ***x1***, ***y1***, ***x2*** and ***y2*** (-1,000≤***x1***,***y1***,***x2***,***y2***≤1,000). These describe a line passing through (***x1***,***y1***) and (***x2***,***y2***). Note that we're interested in the whole, infinite line, not just the segment between (***x1***,***y1***) and (***x2***,***y2***). In any test case, no more than two lines will intersect at any point inside the circle, no lines will be tangent to the circle, and no two lines will be coincident. The input will end with a line with four **0**s.

Output

For each test case, output a single integer, indicating the number of parts that the circle is cut into. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
16 1 7 4 -15 -9 14 -11 -4 30 -3 -20 -20 12 -10 7 17 10 31 0 0 0 0 0	5

B: Nested Palindromes

Palindromes are numbers that read the same forwards and backwards. Your friend Percy recently became interested in a special kind of palindrome that he calls a *Nested Palindrome*. A *Nested Palindrome* meets three conditions:

- The number is a palindrome.
- Split the number in the middle. The first half of the digits of the number is also a *Nested Palindrome*. If the number has an odd number of digits, don't consider the middle digit as part of the first half.
- No two adjacent digits are the same.

Percy says that he has written a *Nested Palindrome* with no leading zeros on a slip of paper. Next, Percy says that he has erased some of the digits in the number and replaced those digits with question marks. He asks you to think about all possible numbers, in increasing order, that can fill those digits and could possibly form the number Percy wrote. Of course, Percy may not be telling the truth about having written a *Nested Palindrome* in the first place.

Percy tells you that the number he wrote is the ***k***th number of this potentially large list. Your task is to find that ***k***th number.

Input

There will be several test cases in the input. Each test case will consist of two lines. The first line will contain an integer ***k*** ($1 \le k \le 10^{18}$), which is the position in the ordered list you must find. The second line contains a string of length 1 to 10,000, consisting only of digits ('0' to '9') and question marks ('?'). Input is terminated by a line with a single **0**.

Output

For each test case, output the *Nested Palindrome* that Percy is looking for. If that number does not exist, or if the string cannot form a *Nested Palindrome*, output **-1**. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
1 1?1 1 ?3? 1 ?1? 55 ??? 55 1?1 3 0?0 0	101 131 212 707 -1 -1

C: Ping!

Suppose you are tracking some satellites. Each satellite broadcasts a 'Ping' at a regular interval, and the intervals are unique (that is, no two satellites ping at the same interval). You need to know which satellites you can hear from your current position. The problem is that the pings cancel each other out. If an even number of satellites ping at a given time, you won't hear anything, and if an odd number ping at a given time, it sounds like a single ping. All of the satellites ping at time 0, and then each pings regularly at its unique interval.

Given a sequence of pings and non-pings, starting at time 0, which satellites can you determine that you can hear from where you are? The sequence you're given may, or may not, be long enough to include all of the satellites' ping intervals. There may be satellites that ping at time 0, but the sequence isn't long enough for you to hear their next ping. You don't have enough information to report about these satellites. Just report about the ones with an interval short enough to be in the sequence of pings.

Input

There will be several test cases in the input. Each test case will consist of a single string on its own line, with from 2 to 1,000 characters. The first character represents time 0, the next represents time 1, and so on. Each character will either be a 0 or a 1, indicating whether or not a ping can be heard at that time (0=No, 1=Yes). Each input is guaranteed to have at least one satellite that can be heard. The input will end with a line with a single **0**.

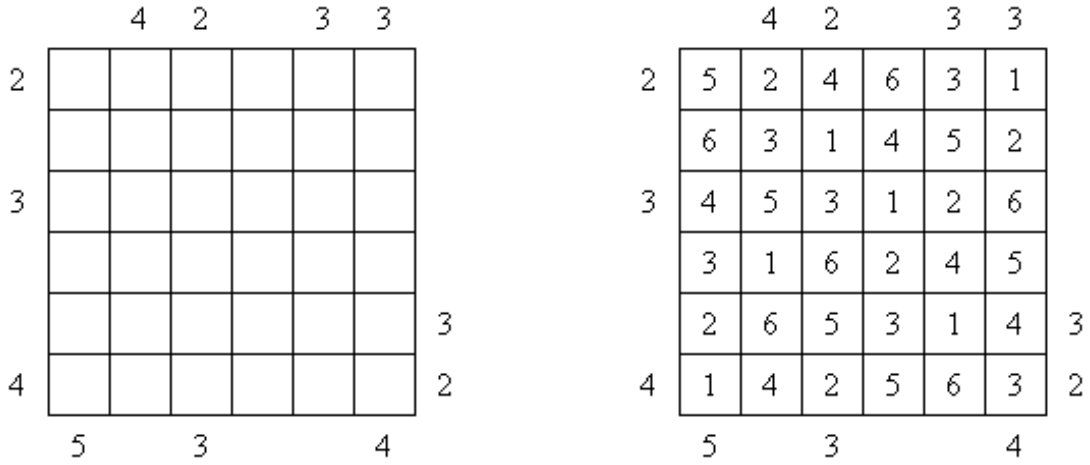
Output

For each test case, output a list of integers on a single line, indicating the intervals of the satellites that you know you can hear. Output the intervals in order from smallest to largest, with a single space between them. Output no extra spaces, and do not separate answers with blank lines.

Sample Input	**Sample Output**
01000101101000	1 2 3 6 8 10 11 13
1001000101001000	3 6 7 12 14 15
0	

D: Skyscrapers

Skyscrapers is a pencil puzzle. It's played on a square ***n***x***n*** grid. Each cell of the grid has a building. Each row, and each column, of the grid must have exactly one building of height 1, height 2, height 3, and so on, up to height ***n***. There may be numbers at the beginning and end of each row, and each column. They indicate how many buildings can be seen from that vantage point, where taller buildings obscure shorter buildings. In the game, you are given the numbers along the outside of the grid, and you must determine the heights of the buildings in each cell of the grid.



Consider a single row of a puzzle of size ***n***x***n***. If we know how many buildings can be seen from the left, and from the right, of the row, how many different ways are there of populating that row with buildings of heights 1..***n***?

Input

There will be several test cases in the input. Each test case consists of three integers n a single line: **n** (1≤***n***≤5,000), ***left*** (1≤***left***≤***n***), and ***right*** (1≤***right***≤***n***), where ***n*** is the size of the row, and ***left*** and ***right*** are the number of buildings that can be seen from the left and right, respectively. The Input will end with a line with three **0**s.

Output

For each test case, print a single integer indicating the number of permutations which satisfy the constraints, modulo 1,000,000,007 (that's not a misprint, the last digit is a seven). Output no extra spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
3 2 2 4 1 2 0 0 0	2 2

E: Count your Cousins

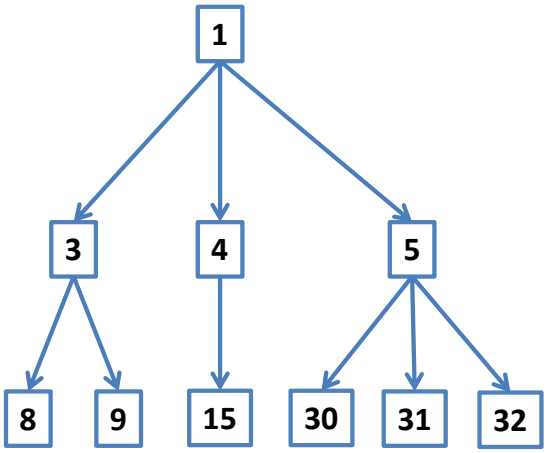
A tree is formed from a strictly increasing sequence of integers as follows:

- The first integer in the sequence is the root of the tree
- The next set of consecutive integers in the sequence describes the children of the root. The first of these will be greater than ***root***+1.
- From there, each set of consecutive integers describes the children of the lowest numbered node which does not yet have children.
- Non-consecutive integers mark a break between one set of children and the next.

For example, the sequence:

```
1 3 4 5 8 9 15 30 31 32
```

Would produce the following tree:



Two nodes are considered to be *Cousins* if they have different parents, but their parents are siblings. Given a tree and a particular node of that tree, count the number of *Cousins* of the node.

Input

There will be several test cases in the input. Each test case will begin with a line with two integers, ***n*** (1≤***n***≤1,000) and ***k*** (1≤***k***≤1,000,000), where ***n*** is the number of nodes in the tree, and ***k*** is the particular node of interest. On the following line will be ***n*** integers, all in the range from 1 to 1,000,000, and guaranteed to be strictly increasing. These describe the tree, in the manner described above. The integers will be separated with a single space. There will be no extra spaces. The value ***k*** is guaranteed to be one of the integers on the second line. Input will end with a line with two **0**s.

Output

For each test case, output a single integer, indicating the number of cousins of node ***k***. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
10 15 1 3 4 5 8 9 15 30 31 32 12 9 3 5 6 8 9 10 13 15 16 22 23 25 10 4 1 3 4 5 8 9 15 30 31 32 0 0	5 1 0

F: Decimal Representation

Any rational fraction can be represented in decimal notation. Using parentheses to denote repeating decimal digits, consider the following examples:

```
4/2 = 2
1/4 = 0.25
10/3 = 3.(3)
1/7 = 0.(142857)
1/45 = 0.0(2)
```

Each of these requires a different number of characters. 4/2 = 2, so it requires only one character. 1/7 = 0.(142857), so it needs 10. Given an integer ***n***, what is the greatest number of characters needed to represent any fraction **a**/**b**, where 1≤**a**,**b**≤***n***?

Input

There will be several test cases in the input. Each test case will consist of a single integer ***n*** (1≤***n***≤500) on its own line. Input will end with a line with a single **0**.

Output

For each test case, output a single integer, indicating the maximum number of characters needed to represent any **a**/**b**, where 1≤**a**,**b**≤***n***. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
12	10
19	22
156	152
0	

G: Tandem Repeats

Tandem Repeats occur in DNA when a pattern of one or more nucleotides is repeated, and the repetitions are directly adjacent to each other. For example, consider the sequence:

ATTCGATTCGATTCG

This contains 9 *Tandem Repeats*:

ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG
ATTCGATTCGATTCG

Given a nucleotide sequence, how many *Tandem Repeats* occur in it?

Input

There will be several test cases in the input. Each test case will consist of a single string on its own line, with from 1 to 100,000 capital letters, consisting only of **A**, **G**, **T** and **C**. This represents a nucleotide sequence. The input will end with a line with a single **0**.

Output

For each test case, output a single integer on its own line, indicating the number *Tandem Repeats* in the nucleotide sequence. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
AGGA	1
AGAG	1
ATTCGATTCGATTCG	9
0	

H: Triangles

Given ***n*** points in a plane, find the triangles with the smallest and largest areas formed by any three of the points.

Input

There will be several test cases in the input. Each test case will begin with an integer ***n*** ($3 \le n \le 2{,}000$) on its own line, indicating the number of points. On each of the next ***n*** lines will be two integers ***x*** and ***y*** ($-10{,}000 \le x, y \le 10{,}000$), representing a point. No test case will contain duplicate points. The input will end with a **0** on its own line.

Output

For each case, output the areas of the smallest, then largest, triangles formed by any 3 of the points in the test case. Output these numbers with exactly one decimal place of accuracy, with exactly one space between them. Output no extra spaces, and do not separate answers with blank lines.

Sample Input

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

Sample Output

```
10.5 33.0
0.0 4.0
```

I: It Takes a Village

As a Sociologist, you are studying a certain kingdom. This kingdom has a capitol city, several villages, and roads between them all. Through your sociological studies, you have been able to determine that there are three separate conditions under which one village will economically affect another village. Village P will affect village Q if ANY of the following are true:

1. If there are two completely different paths to get from village P to village Q, with no villages in common (other than P and Q).
2. If every path from Q to the capitol goes through P.
3. If P affects village R and R affects Q.

The kingdom is starting to build trading posts, to boost the economic health of its villages. When it builds a trading post, it increases the overall revenue of the village it is placed in, and of all villages which are affected by that village according to the above rules. Now, the king wants to know the effect of his new trading posts, so he occasionally asks you to tell him the revenue of a certain village.

Given a sequence of the kings actions, both building trading posts and asking about a certain village, answer his questions.

Input

There will be several test cases in the input. Each test case will begin with a line with three integers, ***n*** (1≤***n***≤100,000), ***m*** (0≤***m***≤100,000), and ***q*** (1≤***q***≤200,000), where ***n*** is the number of villages, ***m*** is the number of roads, and ***q*** is the number of actions the king performs. The villages are numbered **1**..***n***, and **1** represents the capitol.

On each of the next ***m*** lines will be a two integers ***a***, ***b*** (1≤***a***,***b***≤***n***, ***a***!=***b***), representing a road from village ***a*** to village ***b***. The roads are two-way, supporting travel in either direction. It is possible to get from the capitol to every village by some route.

The next ***q*** lines represent the king's actions, in order, and will have one of two forms:

`+` ***k x***

Here, the king builds a trading post at village ***k*** (1≤***k***≤***n***), which increases all affected villages' revenues by ***x*** (1≤***x***≤1,000).

`?` ***k***

Here, the king asks you for the total revenue for village ***k*** (1≤***k***≤***n***), including any trading posts in that village, and all villages that affect that village.

The parts of the king's commands will be separated by a single space, and will have no leading or trailing blanks. The input will end with a line with two **0**s.

Output

For each '? k' question the king asks, print a single integer in its own line, representing the answer to that question. Answer the king's questions in order. Output no spaces, and do not separate answers with blank lines.

Sample Input

```
3 2 4
1 2
3 1
+ 1 1
? 3
+ 2 3
? 3
2 2 4
1 2
2 1
+ 1 1
? 2
+ 2 3
? 1
0 0 0
```

Sample Output

```
1
1
1
4
```

J: You Win!

You just achieved the High Score on your favorite video game! Now, you get to enter your name! You have to use the controller to enter your name, which can be awkward. Here's how it works:

- There are only the 26 capital letters **A** to **Z**, in order. There are no numbers, spaces, lower case letters, or any other characters.
- Pushing UP or DOWN changes the active letter one letter forward (UP) or backward (DOWN). The active letter starts at **A**. It will not reset when you move around in the name. It also wraps: UP from **Z** goes to **A**, DOWN from **A** goes to **Z**.
- Pushing LEFT or RIGHT moves the cursor one letter left or right in the current name. Note that once the cursor is at either end of the current name, it cannot move any further in that direction.
- Pushing the FIRE button adds the active letter to the name.

For example, consider the name 'ALMA'. One way you could enter 'ALMA' is like this:

Action	# of Pushes	Name (| = Cursor)	Active Letter
FIRE	1	A|	A
UP	11	A|	L
FIRE	1	AL|	L
UP	1	AL|	M
FIRE	1	ALM|	M
DOWN	12	ALM|	A
FIRE	1	ALMA|	A

This would take 28 button pushes. However, consider entering 'ALMA' like this:

Action	# of Pushes	Name (| = Cursor)	Active Letter
FIRE	1	A|	A
FIRE	1	AA|	A
LEFT	1	A|A	A
UP	11	A|A	L
FIRE	1	AL|A	L
UP	1	AL|A	M
FIRE	1	ALM|A	M

This takes only 17 button pushes. Given a name, what is the fewest number of button pushes needed to enter that name? Assume that the active letter starts at **A**, and that it doesn't matter where the cursor ends up when you're done.

Input

There will be several test cases in the input. Each test case will consist of a single string on its own line, with from 1 to 18 capital letters, representing a name that must be entered into the High Score list. The input will end with a line with a single **0**.

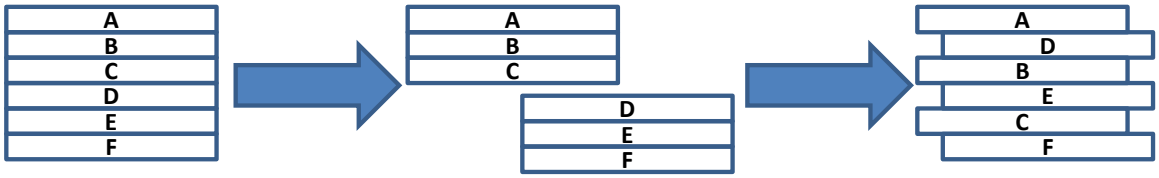
Output

For each test case, output a single integer representing the smallest number of button pushes needed to enter the name. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
ALMA YES 0	17 21

K: Perfect Shuffle

A *Perfect Shuffle* of a deck of cards is executed by dividing the deck exactly in half, and then alternating cards from the two halves, starting with the top half.



Given a deck of cards, perform a *Perfect Shuffle*. If there is an odd number of cards, give the top half split one more card than the bottom half.

Input

There will be several test cases in the input. Each test case will begin with a line with a single integer ***n*** (1≤***n***≤1,000), indicating the number of cards. On each of the next ***n*** lines will be a string from 1 to 80 characters in length, which is the name of a card. It will contain only capital letters and dashes. Within a test case, all card names will be unique. Input will end with a line with a single **0**.

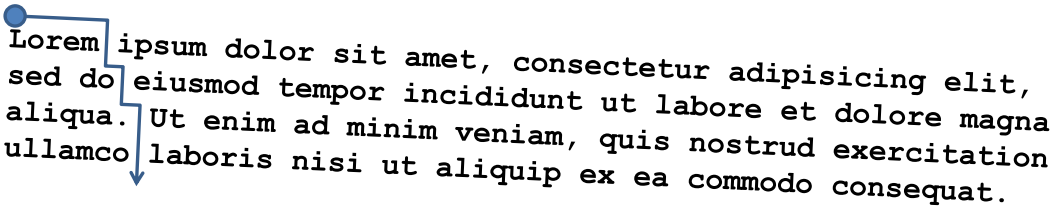
Output

For each test case, output ***n*** lines, consisting of the deck after a perfect shuffle. Output no extra spaces. Do not print a blank line between answers.

Sample Input	Sample Output
4 ACE KING QUEEN JACK 5 SKIP DRAW-TWO REVERSE WILD WILD-DRAW-FOUR 0	ACE QUEEN KING JACK SKIP WILD DRAW-TWO WILD-DRAW-FOUR REVERSE

L: Text Roll

Take some text. Put a small ball at the top of the first letter of the first word of the first sentence. The ball is drawn via gravity downwards. The text is also at a slight angle, so the ball wants to also move towards the right. The ball can freely move between the lines, and can drop through spaces. Considering the first column to be column 1, what column will the ball end up in? In this example, the ball ends up in column 8.



Input

There will be several test cases in the input. Each test case will begin with an integer ***n*** ($1 \le n \le 1{,}000$) on its own line, indicating the number of lines of text. On each of the next ***n*** lines will be text, consisting of printable ASCII characters and spaces. There will be no tabs, nor any other unprintable characters. Each line will be between 1 and 100 characters long. The input will end with a line containing a single **0**.

Output

For each test case, output a single integer on its own line, indicating the column from which the ball will drop. Output no spaces, and do not separate answers with blank lines.

Sample Input

```
4
Lorem ipsum dolor sit amet, consectetur adipisicing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.
3
Supercalifragilisticexpialidocious! Can you handle
short
lines?
0
```

Sample Output

```
8
36
```