A

Attacking rooks

Chess inspired problems are a common source of exercises in algorithms classes. Starting with the well known 8-queens problem, several generalizations and variations were made. One of them is the N-rooks problem, which consists of placing N rooks in an N by N chessboard in such a way that they do not attack each other.

Professor Anand presented the N-rooks problem to his students. Since rooks only attack each other when they share a row or column, they soon discovered that the problem can be easily solved by placing the rooks along a main diagonal of the board. So, the professor decided to complicate the problem by adding some pawns to the board. In a board with pawns, two rooks attack each other if and only if they share a row or column and there is no pawn placed between them. Besides, pawns occupy some squares, which gives an additional restriction on which squares the rooks may be placed on.

Given the size of the board and the location of the pawns, tell Professor Anand the maximum number of rooks that can be placed on empty squares such that no two of them attack each other.

Input

The first line contains an integer N ($1 \leq N \leq 100$) representing the number of rows and columns of the board. Each of the next N lines contains a string of N characters. In the i-th of these strings, the j-th character represents the square in the i-th row and j-th column of the board. The character is either "`.`" (dot) or the uppercase letter "`X`", indicating respectively an empty square or a square containing a pawn.

Output

Output a line with an integer representing the maximum number of rooks that can be placed on the empty squares of the board without attacking each other.

Sample input 1	Sample output 1
5 X.... X.... ..X.. .X... X	7

Sample input 2	Sample output 2
4 .... .X..	5

Sample input 3	Sample output 3
1 X	0

B

Blogger language

Benjamin's granddaughter Brenda has a blog where she posts articles about school, friends and other life issues. Intrigued by her opinions, Benjamin tried to read it, but very soon he realized it was too hard to read because of Brenda's writing quirks.

Brenda writes without spaces or punctuation marks, and moreover, she uses lower and uppercase letters in a liberal and strange way. For example, one of her posts is "`PrOgRAMmINgiSgrEAt`". Benjamin has trouble noticing the words "`programming`", "`is`" and "`great`" when they are written in this way.

To improve his understanding Benjamin decided to do the following: he will first choose a particular string T and a blog post he is interested in; then he will select a contiguous substring of the post and search for T within the substring, in a case-insensitive way; for each occurrence of T within the substring, he will calculate the number of case mismatches, and finally he will obtain the maximum among all these values. For example, if Benjamin chooses "`GR`" as T and then selects the substring "`PrOgRAM`", he would find a single occurrence "`gR`" for which the number of case mismatches is 1. For the same substring, if "`r`" was chosen as T, he would have found two occurrences, "`r`" with 0 mismatches and "`R`" with 1 mismatch, so the maximum number of mismatches would be 1.

To complicate things further, Brenda included in the blog a script that, after operating with a substring selection, flips the case of all the selected letters. This means that after selecting "`PrOgRAM`" and proceeding as explained above, the sample post would read "`pRoGrammINgiSgrEAt`". If Benjamin selects "`ammINgi`" as a second substring, after calculating his result the post would be left as "`pRoGrAMMinGISgrEAt`", accumulating both flips.

You will be given the string T and the original text of the blog post chosen by Benjamin. You will also be given a list of substring selections Benjamin made, in the order he made them. You need to calculate, for each selection, the maximum number of case mismatches of the occurrences of T in the selected part, considering all the case flips made by previous selections. Notice that the flipping of the case occurs after calculating the result for each selection.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$) and a non-empty string T of at most 5 letters, representing respectively the number of substring selections and the string to search for. The second line contains a non-empty string P of at most 10^5 letters, indicating the original text of the blog post. Positions of the post are numbered with consecutive integers from left to right, being 1 the leftmost position and $|P|$ the rightmost position. Each of the next N lines describes a substring selection with two integers L and R ($1 \leq L \leq R \leq |P|$) indicating that the substring starts at position L and ends at position R, inclusive.

Output

Output N lines, each of them containing an integer. In the i-th line write the maximum number of case mismatches of the occurrences of T in the i-th substring selection, considering all the case flips made by previous selections; if no such occurrence exists write the value -1.

Sample input 1	**Sample output 1**
3 gR PrOgRAMmINgiSgrEAt 1 7 4 18 6 14	0 2 -1

Sample input 2	**Sample output 2**

`9 abCAb`
`aBcAbCAbaBCAb`
`1 13`
`1 13`
`4 8`
`5 11`
`3 11`
`4 10`
`1 13`
`8 8`
`1 13`
	`2`
	`4`
	`1`
	`-1`
	`0`
	`5`
	`2`
	`-1`
	`4`

C
Football

Your favorite football team is playing a charity tournament, which is part of a worldwide fundraising effort to help children with disabilities. As in a normal tournament, three points are awarded to the team winning a match, with no points to the losing team. If the game is drawn, each team receives one point.

Your team played N matches during the first phase of the tournament, which has just finished. Only some teams, the ones with more accumulated points, will advance to the second phase of the tournament. However, as the main objective of the tournament is to raise money, before the set of teams that will pass to the second phase is determined, each team is allowed to buy additional goals. These new goals count as normally scored goals, and may be used to alter the result of any of the matches the team played.

Your team's budget is enough to buy up to G goals. Can you tell the maximum total number of points your team can get after buying the goals, supposing the other teams will not buy any goals?

Input

The first line contains two integers N ($1 \leq N \leq 10^5$) and G ($0 \leq G \leq 10^6$) representing respectively the number of matches your team played and the number of goals your team can buy. Each of the next N lines describes a match result with two integers S and R ($0 \leq S, R \leq 100$), indicating respectively the goals your team scored and received on that match before buying goals.

Output

Output a line with an integer representing the maximum total number of points your team can get after buying the goals.

Sample input 1	Sample output 1
2 1 1 1 1 1	4

Sample input 2	Sample output 2
3 2 1 3 3 1 2 2	6

Sample input 3	Sample output 3
4 10 1 1 2 2 1 3 0 4	12

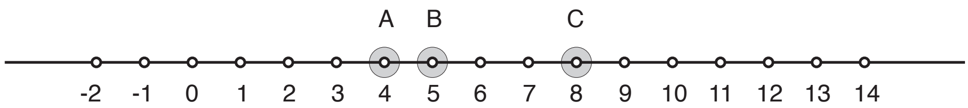
D

Ingenious Metro

Problem code name: `ingenious`

The King of Logonia will inaugurate soon a new and revolutionary metro, based on an invention of the Royal Engineers, which allows teletransportation.

The new metro consists of a very long tunnel with a station at each kilometer. There are also T teletransporters, which are located at some of the stations. In each station there is a keyboard with T keys, where each key corresponds to one teletransporter. The figure below illustrates a metro system with three teletransporters, located in stations marked A, B and C.



The metro works as follows. The user goes in a station (the *start* station) and presses the key corresponding to the teletransporter he wants to use. The user is then teletransported to the station which is at the same distance from the teletransporter as the start station, but on the opposite side relative to the teletransporter. More precisely, if the location of the start station is i and the user presses the key corresponding to the teletransporter located in position j, he will be taken to the station located at position $2 \times j - i$. For example, if the user is in station 6 and wants to go to station -2, he can use the teletransporter C (goes from 6 to 10) and then the teletransporter A (goes from 10 to -2).

The King, however, knows that it is possible that there is no sequence of teletransporters that will take the user from a given station X to a given station Y. To avoid that the users keep trying to go where they cannot go, he wants to make a program available in the Internet to help users. The King wants you to write a program which, given the position of each teletransporter, answers a series of queries. For each query the start and the destination stations are given, and your program must determine if it is possible for the user to go from start to destination.

Input

Each test case is given using several lines. The first line contains two integers T and Q indicating respectively the number of teletransporters ($1 \leq T \leq 10^5$) and the number of queries ($1 \leq Q \leq 10$). The second line contains T different integers t_i indicating the position of the teletransporters ($-10^7 \leq t_i \leq 10^7$). Each of the Q following lines describes a query and contains two distinct integers S and D indicating the position of the start and destination stations ($-10^7 \leq S, D \leq 10^7$).

The last test case is followed by a line containing two zeros.

Output

For each test case output a single line containing the answers to the Q queries, in the same order that the queries were given in the input. For each query you must output an uppercase

'Y' if it is possible to reach the destination station from the start station using the metro, or an uppercase 'N' otherwise.

Sample input	Output for the sample input
1 1 -2 -6 2 5 2 10 20 30 40 50 10 15 20 40 5 3 0 5 -3 -8 4 -1 499 4 237 -1 -591 0 0	Y N Y Y N Y

E

Another Crisis

File code name: `another`

A couple of years ago, a new world wide crisis started, leaving many people with economical problems. Some workers of a particular company are trying to ask for an increase in their salaries.

The company has a strict hierarchy, in which each employee has exactly one direct boss, with the exception of the owner of the company that has no boss. Employees that are not bosses of any other employee are called *workers*. The rest of the employees and the owner are called *bosses*.

To ask for a salary increase, a worker should file a petition to his direct boss. Of course, each boss is encouraged to try to make their subordinates happy with their current income, making the company's profit as high as possible. However, when at least T percent of its direct subordinates have filed a petition, that boss will be pressured and have no choice but to file a petition himself to his own direct boss. Each boss files at most 1 petition to his own direct boss, regardless on how many of his subordinates filed him a petition. A boss only accounts his direct subordinates (the ones that filed him a petition and the ones that didn't) to calculate the pressure percentage.

Note that a boss can have both workers and bosses as direct subordinates at the same time. Such a boss may receive petitions from both kinds of employees, and each direct subordinate, regardless of its kind, will be accounted as 1 when checking the pressure percentage.

When a petition file gets all the way up to the owner of the company, all salaries are increased. The workers' union is desperately trying to make that happen, so they need to convince many workers to file a petition to their direct boss.

Given the company's hierarchy and the parameter T, you have to find out the minimum number of workers that have to file a petition in order to make the owner receive a petition.

Input

There are several test cases. The input for each test case is given in exactly two lines. The first line contains two integers N and T ($1 \leq N \leq 10^5, 1 \leq T \leq 100$), separated by a single space. N indicates the number of employees of the company (not counting the owner) and T is the parameter described above. Each of the employees is identified by an integer between 1 and N. The owner is identified by the number 0. The second line contains a list of integers separated by single spaces. The integer B_i, at position i on this list (starting from 1), indicates the identification of the direct boss of employee i ($0 \leq B_i \leq i-1$).

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

Output

For each test case output a single line containing a single integer with the minimum number of workers that need to file a petition in order to get the owner of the company to receive a petition.

Sample input	Output for the sample input
3 100 0 0 0 3 50 0 0 0 14 60 0 0 1 1 2 2 2 5 7 5 7 5 7 5 0 0	3 2 5

F

File Recover

File code name: `file`

Your school has a computer that is used as a web server for hosting its institutional web site, personal pages of the staff, sites for research groups, subjects, and many others.

Recently, the hard disk table was corrupted, so the organization of all the files was lost. Sadly enough, there are no backups of that information. The only hope is to look through the entire disk data and try to find out which parts correspond to each file. Fortunately, the disk was using a file system that kept each individual file contiguous, so only contiguous pieces of data need to be inspected.

The disk data is a sequence of bytes. Each byte in this particular disk can store an English alphabet letter (distinguishing lowercase and uppercase), a decimal digit, a point or a comma, making a total of 64 different characters.

While you were thinking about how to solve the problem, you suddenly remembered that the file system also maintained multiple copies of each file, so only the pieces of contiguous bytes that are repeated had a chance of being a file. Moreover, for each repetition of the same contiguous bytes, only one copy needs to be checked. For instance, if the data is '*ababcabb*', the contiguous subsequences '*a*', '*b*' and '*ab*' are repeated, but nothing containing '*c*', nor '*ba*' or '*bb*' is. Therefore, we have 3 pieces of contiguous bytes that need checking in this case.

You decide to write a program that computes exactly how many sequences need checking, that is, the number of different sequences of contiguous bytes that appear at least twice in the data.

Input

There are several test cases. The input of each test case is given in exactly one line, containing a non-empty string of at most 10^5 characters that represents the disk data. Each character in the string is either a lowercase letter, an uppercase letter, a digit, a point or a comma.

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

Output

For each test case output a single line with an integer, representing the number of different contiguous subsequences that appear at least twice in the input string.

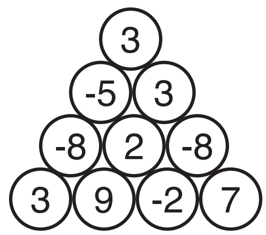
Sample input	Output for the sample input
`ababcabb`	`3`
`mississippi`	`9`
`aaaaaaaaaaaaaaaaaaaaaaaaa`	`25`
`012345678,abcdefg.STUVWXYZ`	`0`
`say.twice,say.twice`	`45`
`*`	

G

Ball Stacking

Problem code name: `ball`

The XYZ TV channel is developing a new game show, where a contestant has to make some choices in order to get a prize. The game consists of a triangular stack of balls, each of them having an integer value, as the following example shows.



The contestant must choose which balls he is going to take and his prize is the sum of the values of those balls. However, the contestant can take any given ball only if he also takes the balls directly on top of it. This may require taking additional balls using the same rule. Notice that the contestant may choose not to take any ball, in which case the prize is zero.

The TV show director is concerned about the maximum prize a contestant can make for a given stack. Since he is your boss and he does not know how to answer this question, he assigned this task to you.

Input

Each test case is described using several lines. The first line contains an integer N representing the number of rows of the stack ($1 \leq N \leq 1000$). The i-th of the next N lines contains i integers B_{ij} ($-10^5 \leq B_{ij} \leq 10^5$ for $1 \leq j \leq i \leq N$); the number B_{ij} is the value of the j-th ball in the i-th row of the stack (the first row is the topmost one, and within each row the first ball if the leftmost one).

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

Output

For each test case output a line with an integer representing the maximum prize a contestant can make from the stack.

Sample input	Output for the sample input
4 3 -5 3 -8 2 -8 3 9 -2 7 2 -2 1 -10 3 1 -5 3 6 -4 1 0	7 0 6

H

Different Digits

The inhabitants of Nlogonia are very superstitious. One of their beliefs is that street house numbers that have a repeated digit bring bad luck for the residents. Therefore, they would never live in a house which has a street number like 838 or 1004.

The Queen of Nlogonia ordered a new seaside avenue to be built, and wants to assign to the new houses only numbers without repeated digits, to avoid discomfort among her subjects. You have been appointed by Her Majesty to write a program that, given two integers N and M, determines the maximum number of houses that can be assigned street numbers between N and M, inclusive, that do not have repeated digits.

Input

Each test case is described using one line. The line contains two integers N and M, as described above ($1 \leq N \leq M \leq 5000$).

Output

For each test case output a line with an integer representing the number of street house numbers between N and M, inclusive, with no repeated digits.

Sample input	Output for the sample input

`87 104`
`989 1022`
`22 25`
`1234 1234`
	`14`
	`0`
	`3`
	`1`