Problem A. Strong parentheses sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

A sequence is called a correct parenthesis sequence if it's the following form:

- Empty sequence is considered a correct parenthesis sequence.

- (A) is considered a correct parenthesis sequence.

- XY is considered a correct parenthesis sequence, if both X and Y are correct parenthesis sequences.

A sequence is called a strong parenthesis sequence only if it on form (A), where A is a correct parenthesis sequence.

You have to calculate a weird sum: take $[i_1, j_1]$ and $[i_2, j_2]$ every two subarrays of A. The subarrays must not intersect (i.e. $i_1 \leq i_2$ and $j_1 \leq i_2$) and must be strong parenthesis sequences. Then, we add to the sum the minimum length of a subarray, such as it is also a strong parenthesis sequence and it contains both $[i_1, j_1]$ and $[i_2, j_2]$ (i.e. if a subsequence of minimum length is $[i_3, j_3]$, then $[i_3, j_3]$ is a strong parenthesis sequence and also $i_3 \leq i_1 \leq j_1 \leq j_3$ and $i_3 \leq i_2 \leq j_2 \leq j_3$).

Output the calculated sum.

Input

The input contains the sequence A – a strong parenthesis sequence. It's length is between 2 and 100000.

Output

The output contains a single line - the sum requested by the problem.

Examples

standard input	standard output
`(()())`	`6`
`(()(()))`	`16`

Note

In the first test case the answer is length of "`([][])`".

In the second test case the answer is the sum of lenghts of "`([][()])`"and "`([]([]))`".

(The strong parentheses sequences $[i_1, j_1]$ and $[i_2, j_2]$ emphasized by "[...]").

Problem B. Retrospective Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Retrospective sequence is a recursive sequence that is defined through itself. For example Fibonacci specifies the rate at which a population of rabbits reproduces and it can be generalized to a retrospective sequence. In this problem you will have to find the n-th Retrospective Sequence modulo MOD = 1000000009.

The first ($1 \le N \le 20$) elements of the sequence are specified. The remaining elements of the sequence depend on some of the previous N elements.

Formally, the sequence can be written as $F_m = F_{m-k_1} + F_{m-k_2} + ... + F_{m-k_i} + ... + F_{m-k_{C-1}} + F_{m-k_C}$. Here, C is the number of previous elements the m-th element depends on, $1 \le k_i \le N$.

Input

The first line of each test case contains 3 numbers, the number ($1 \le N \le 20$) of elements of the retrospective sequence that are specified, the index ($1 \le M \le 10^{18}$) of the sequence element that has to be found modulo MOD, the number ($1 \le C \le N$) of previous elements the i-th element of the sequence depends on.

The second line of each test case contains N integers specifying $0 \le F_i \le 10$, ($1 \le i \le N$).

The third line of each test case contains $C \ge 1$ integers specifying $k_1, k_2, ..., k_{C-1}, k_C$ ($1 \le k_i \le N$).

Output

Output single integer R, where R is F_M modulo MOD.

Examples

standard input	standard output

2 2 2
1 1
1 2
	1

2 7 2
1 1
1 2
	13

3 100000000000 3
0 1 2
1 2 3
	48407255

Problem C. Power

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Arthur, the superhero of our times, faces a confusing problem.

He almost completed his final mission. Only one step is left - to find the amulet of power. Well, Arthur knows where the amulet is. But it is not so easy to take him. The reason is that the powerful guard kills anybody who tries to steal the amulet. Unfortunately, the guard is much more powerful than Arthur. So if Arthur meets the guard, the end is not too good.

It is known that the amulet of power can not exist for a long time without the owner. So Arthur has to rush.

You are given the matrix which describes the forest. Arthur and the guard can not leave the forest. The forest contains some dangerous animals. For example, snakes or hungry wolfs. No doubt that Arthur and the guard are afraid of these animals.

Assume that Artur's position is $(r_A; c_A)$ and the guard's position is $(r_G; c_G)$. r, c describes the row and the column of the matrix. Let's define the guard's moves formally:

if $c_A < c_G$, the guard moves $c_G := c_G - 1$

if $c_G < c_A$, the guard moves $c_G := c_G + 1$

if the guard did not make the previous move ($c_A = c_G$ or because of dangerous animals) then he moves with the respect to these rules (if it is possible):

if $r_A < r_G$, the guard moves $r_G := r_G - 1$

if $r_G < r_A$, the guard moves $r_G := r_G + 1$

Arthur can move to one of four adjacent cells (row up, row down, column left, column right). Both, Arthur and the guard can't move diagonally.

Arthur and the guard move alternatively. The first move is made by Arthur. Arthur and the guard make the same number of moves. That is, if Arthur gets the amulet and he is caught during the next guard's move, then Arthur is killed.

Given the description of the forest determine the minimal number of Arthur's steps needed to be made in order to reach the amulet of power.

Input

The first line of each test case contains the number of rows N and the number of columns M in the matrix. $(2 \leq N, M \leq 25)$

The following N lines contains M characters. The j-th character on the i-th line describes the position (i, j) in the matrix.

'A' denotes the Arthur's initial position.

'G' denotes the guard's initial position.

'X' denotes the position occupied by dangerous animals.

'P' denotes the amulet of power.

'.' denotes other cells.

Output

Output single integer R – the minimal numbers of steps needed to be made. If the amulet can not be reached or the Arhur will certainly be caught by the guard, output -1.

Examples

standard input	standard output
2 2 A. GP	-1
2 3 A.. GXP	3
2 2 .G AP	-1

Problem D. Triangle Formation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

You are given N wooden sticks. Your task is to determine how many triangles can be made from the given sticks without breaking them. Each stick can be used in at most one triangle.

Input

The first line of each test case contains the number of sticks N. $(1 \leq N \leq 15)$

The second line of each test case contains N integers len_i that are the lengths of the sticks. $(1 \leq len_i \leq 10^9)$.

Output

Output single integer R – the maximal number of triangles.

Examples

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

Problem E. Restore

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Given a matrix A of size $N * N$. The rows are numbered from 0 to N-1, the columns are numbered from 0 to N-1. In this matrix, the sums of each row, the sums of each column, and the sum of the two diagonals are equal.

For example, a matrix with $N = 3$:

$$\begin{matrix} 2 & 9 & 4 \\ 7 & 5 & 3 \\ 6 & 1 & 8 \end{matrix}$$

The sums of each row:

$2 + 9 + 4 = 15$

$7 + 5 + 3 = 15$

$6 + 1 + 8 = 15$

The sums of each column:

$2 + 7 + 6 = 15$

$9 + 5 + 1 = 15$

$4 + 3 + 8 = 15$

The sums of each diagonal:

$2 + 5 + 8 = 15$

$4 + 5 + 6 = 15$

As you can notice, all sums are equal to 15.

However, all the numbers in the main diagonal (the main diagonal consists of cells (i, i)) have been removed. Your task is to recover these cells.

Input

The first line contains the dimension of the matrix, n ($1 \leq n \leq 100$).

The following n lines contain n integers each, with removed numbers denoted by 0, and all other numbers $-10^{12} \leq A_{ij} \leq 10^{12}$.

Output

The restored matrix should be outputted in a similar format, with n lines consisting of n integers each.

Examples

`standard input`	`standard output`

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

Problem F. Ellipses

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 megabytes

Informikas has married recently, and although the quality of his life has improved since then, his wife does not approve some of his mathematical games which he plays on his own table!

The table has an elliptical form, which means that it is very similar to a circle, and there are no corners, therefore, Informikas's future children will not accidentally hit their heads to a corner of a table. This would hurt and Informikas knows that pain is a terrible emotion, and it is better to avoid it at all. As a result, Informikas is completely satisfied with the form of the table. However, the form of the table is unimportant in this problem (Informikas just wanted to brag about his table).

What Informikas does on his table is more important, which usually is playing with a lot of ellipses, conveniently numbered from 1 to n. Although some of his ellipses might disappear (and you will soon learn why), they are never renumbered. One elliptical figure is characterized by half of the length of its major axis, and half of the length of its minor axis. We will refer to these numbers as a and b.

Similar numbers A and B correspond to a and b, and they are called 'wife's tolerance numbers'. If $a_i > A$ or $b_i > B$, the wife starts to think that the i-th ellipse takes up too much space, and she immediately disposes it to a recycle bin! As opposite to this, if $a_i \le 0$ or $b_i \le 0$, the ellipse just disappears from the table and never reappears again.

You have to process those queries:

Type 1: given l, r and v, you need to set $a_i = v$, for all i in $[l;\ r]$.

Type 2: given l, r and v, you need to set $b_i = v$, for all i in $[l;\ r]$.

Type 3: given l, r and v, you need to set $a_i = a_i + v$, for all i in $[l;\ r]$.

Type 4: given l, r and v, you need to set $b_i = b_i + v$, for all i in $[l;\ r]$.

Type 5: given l and r, you need to print the sum of the areas of ellipses numbered from l to r, divided by π. If an ellipse in the interval does not lay on the table, its area is assumed to be 0.

Type 6: given l and r, you need to print the number of ellipses numbered from l to r, which still lay on the table.

Input

The first line of the input contains n and m ($1 \le n, m \le 10^5$) - the number of ellipses and the number of queries respectively.

The following line contains $2n$ integers, the i-th pair describes the initial a and b of the i-th ellipse ($1 \le a, b \le 10^6$).

The following m lines describe queries in the format t, l, r and v (if needed) ($1 \le t \le 6$, $1 \le l \le r \le n$, $-10^6 \le v \le 10^6$) - the type of query, the endpoints of the given interval and the modification value. All numbers are integers, and they are separated by spaces.

On the last line of the input there are numbers A and B ($1 \le A, B \le 10^6$).

Output

For each type 5 or type 6 query print one line with the corresponding answer.

Examples

standard input	standard output

1 3
1 1
6 1 1
3 1 1 1
6 1 1
1 1
	1
	0

Problem G. LCS Revised

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

The longest common subsequence is a well known DP problem: given two strings A and B, one has to compute the maximum length of a subsequence that's common to both A and B.

In this particular problem we work with strings A and B formed only by 0 and 1, having the same length. You're given a string A of length n. Iterate all strings B possible. There are 2^n of them. Calculate, for each string B, the longest common subsequence of A and B. Then, output the minimum length obtained.

Input

The first and the only line of the input contains string A, formed only by 0 and 1. It's guaranteed that the length is between 1 and 10^5.

Output

Output a single number - the requested length.

Examples

standard input	standard output
`101010`	`3`

Problem H. Words from cubes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Informikas was cleaning his drawers while he found a toy of his childhood. Well, it's not just a toy, it's a bunch of cubes with letters and digits written on them.

Informikas remembers that he could have make any word he could think of using these cubes. He is not sure about that now, because some of the cubes have been lost.

Informikas has already come up with a word he would like to make. Could you help him by saying if the word can be built from the cubes in the drawer?

Input

On the first line of input there is a string S, consisting of lowercase English letters, and an integer N ($4 \le |S| \le 20$, $1 \le N \le 100$) – the word Informikas want to build and the number of cubes. On the every of the following N lines there are 6 characters. Every of those characters is either a lowercase English letter or a digit.

It is guaranteed that the string S consists only of lowercase English letters.

Output

Output one word, either "YES if the word can be built using given cubes, or "NO"otherwise.

Examples

standard input	standard output

`dogs 4`
`d 1 w e 7 9`
`o 2 h a v e`
`g 3 c o o k`
`s 3 i e s 5`
	`YES`

`banana 6`
`b a 7 8 9 1`
`n 1 7 7 7 6`
`a 9 6 3 7 8`
`n 8 2 4 7 9`
`a 7 8 9 1 3`
`s 7 1 1 2 7`
	`NO`

Problem I. Yet another A + B

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

You are given three numbers. Is there a way to replace variables A, B and C with these numbers so the equality $A + B = C$ is correct?

Input

There are three numbers X_1, X_2 and X_3 ($1 \leq X_i \leq 10^{100}$), each on a separate line of input.

Output

Output either "YES if there is a way to substitute variables A, B and C with given numbers so the equality is correct, or "NO"otherwise.

Examples

standard input	standard output
1 2 3	YES
1 2 4	YES
1 3 5	NO