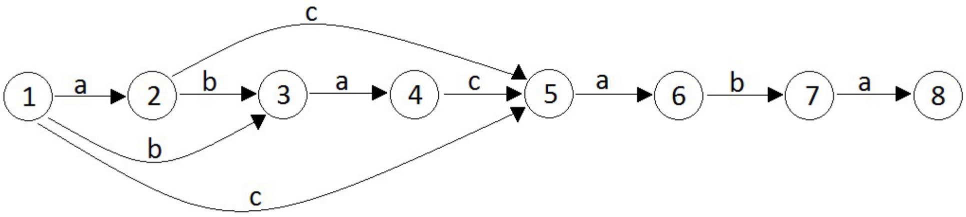
Problem A. Automaton (Division 1 Only!)

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

For a given string S, create determinstic finite-state automaton, that accepts all suffixes of S (and probably other finite strings). Automaton must consist of minimal number of states — N and no more than $2 \cdot N$ transitions.

Image below shows automaton from test sample for string "`abacaba`"



Input

The only line contains string S, consisting of lowercase latin letters.

Output

In the first line print two numbers N and K are amount of states and transitions, followed by K lines, each having two numbers a_i, b_i and letter c_i, meaning transtion from state a_i to b_i by letter c_i. You can output transitions in any order. If there are multiple solutions, output any of them.

Limits

$1 \leqslant |S| \leqslant 10^5$

$1 \leqslant K \leqslant 2N$

$1 \leqslant a_i, b_i \leqslant N$

$'a' \leqslant c_i \leqslant 'z'$

Example

standard input	standard output

`abacaba`	`8 10`
	`1 2 a`
	`1 3 b`
	`1 5 c`
	`2 3 b`
	`2 5 c`
	`3 4 a`
	`4 5 c`
	`5 6 a`
	`6 7 b`
	`7 8 a`

Problem B. Beinz (Division 1 Only!)

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

k-dimensional Pascal's simplex is a k-dimensional figure where cell with coordinates $(i_1, i_2, ..., i_k)$ contains value $C_n^{i_1, i_2, ..., i_k}$, where $i_j \geq 0$, and $n = i_1 + i_2 + ... + i_k$

Example:

1-dimensional Pascal's simplex is infinite line of ones

2-dimensional Pascal's simplex is the usual Pascal's triangle.

3-dimensional Pascal's simplex is an infinite tetrahedron, which consists of all combinations of a kind $C_n^{i,j,k}$ and so on.

Suppose, we have k-dimensional Pascal's simplex, in which all combinations are computed modulo p (p is prime number). Consider nth layer of this simplex (the one where all combinations of n are written). Find the number of non-zero elements of this layer. Since that number can be large, it's necessary to write it modulo $10^9 + 7$.

Input

Three numbers k, p and t are given in the first line. Each of next t lines has one number n, that is number of layer.

Output

Exactly t lines, each containing amount of non-zero elements on nth simplex's layer, modulo $10^9 + 7$ for correspoding n.

Limits

$1 \leqslant k \leqslant 10^3$

$2 \leqslant p \leqslant 10^6 + 3$, p — prime number.

$1 \leqslant t \leqslant 10^5$

$0 \leqslant n \leqslant 10^{18}$

Example

standard input	standard output
2 7 4 0 6 7 8	1 7 2 4
3 7 4 0 6 7 8	1 28 3 9

Problem C. Cutting (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds (*3 seconds for Java*)
Memory limit:	256 mebibytes

You are given string S and list M, consisting of N words. During an operation you may choose substring of string S, if it can be found as word in list M, and cut it out of string S. Then remaining parts of string S are merged if there are any. Determine minimal amount of opertions required to erase whole string S. It is guaranteed that it can be done.

Input

The first line contains word S. Second line contains integer N, amount of words in the list. It is followed by N lines listing words from M. All words consist of lowercase latin letters only.

Output

One integer, that is minimal number of opertions required to erase whole string S.

Limits

$1 \leqslant |S| \leqslant 100$

$1 \leqslant N \leqslant 100$

$1 \leqslant |M_i| \leqslant 100$

Example

standard input	standard output
abacaba 4 aba aca a b	3

Remark to example:

First operation cuts substring “`aca`”, result “`abba`”.

Second operation cuts substring “`b`”, result “`aba`”.

Third operation erases string “`aba`”.

Problem D. Disclosure (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

You are given an acyclic directed graph, consisting of N nodes and K edges. Delete maximum amount of edges, without causing graph's transitive closure to change.

Transitive closure of graph G is a graph G', consisting of all nodes of orgignal graph G and such edges (u, v), that there is a path from node u to v in graph G.

Input

The first line contains two numbers N and K. Then K lines follow, each containing two integers a_i and b_i, descripting directed edge from a_i to b_i. Graph doesn't contain loops, cycles and multiple edges.

Output

Output the every edge that is necessary to remain in graph as the pair of nodes connected by this edge. Edges can be written in any order.

Limits

$1 \leqslant N \leqslant 50000$

$0 \leqslant K \leqslant 50000$

$1 \leqslant a_i, b_i \leqslant N$

Example

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

Problem E. Embedded circles (Division 1 Only!)

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

You are given a table $\{a_{ij}\}$ of R rows and C columns, consisting of digits from '0' to '9'. Respond Q queries of a kind:

$i_k\ j_k\ r_k$, $1 \leqslant k \leqslant Q$

Find the sum of all such elements a_{ij}, that $(i - i_k)^2 + (j - j_k)^2 \leqslant r_k^2$.

Summation area for each query is something like circle with center in cell (i_k, j_k) and radius r_k. Area is defined exactly by the formula above, but we will call it a circle for convenience.

For any two queries the following conditions are satisfied: either their circles have no common table cells, or one of the circles is fully contained in another.

Furthermore, queries are sorted by insertion. That means, that if circles of queries k and l ($k < l$) have common cells, then for any t: $k < t \leqslant l$, circle t is fully contained in the circle of query k. Circles of different queries can coinside.

Input

First line contains two numbers R and C that are table size. Then R lines with C numbers in each (no spaces between numbers) follow. The next line has number Q — amount of querries. Each of the next Q lines describes a querry with three integers: i_k, j_k (central cell row and coloumn numbers) and r_k (circle radius).

Output

Because of large number of queries, output the sum of answers for all queries.

Limits

$1 \leqslant R, C \leqslant 2000$

$'0' \leqslant a_{ij} \leqslant '9'$

$1 \leqslant Q \leqslant 10^6$

$1 + r_k \leqslant i_k \leqslant R - r_k$

$1 + r_k \leqslant j_k \leqslant C - r_k$

$0 \leqslant r_k \leqslant \dfrac{\min(R, C) - 1}{2}$

Example

standard input	standard output
6 6 123456 234567 345678 456789 567890 678901 10 1 1 0 3 3 2 3 2 1 2 2 0 4 2 0 1 3 0 2 3 0 4 3 0 5 5 1 5 5 0	141

Explanation

$141 = 1 + 65 + 20 + 3 + 5 + 3 + 4 + 6 + 25 + 9$

Problem F. False figures (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds (*3 seconds for Java*)
Memory limit:	256 mebibytes

This problem is a test validator for previous problem. The most difficult part is already tested, you have only to test positioning of circles.

Imagine table $\{a_{ij}\}$ of 1000 lines and 1000 coloumns. There are Q queries like:

$i_k\ j_k\ r_k$, $1 \leqslant k \leqslant Q$

Each of them defines region of elements of table $\{a_{ij}\}$ such, that $(i - i_k)^2 + (j - j_k)^2 \leqslant r_k^2$.

For convenience this region is called circle with centre in cell (i_k, j_k) and radius r_k.

Pair of queries k and l ($k < l$) is invalid, if circles k and l have common cells and there is circle t: $k < t \leqslant l$, that isn't contained in circle k. Circles of different queries can match.

Determine, if there is at least one invalid pair among queries.

Input

The first line has number Q, that is amount of queries. Then Q lines describe queries with 3 integers in each string i_k, j_k, r_k, which are numbers of line and coloumn of central cell and circle radius.

Output

If there is a pair of invalid queries, write numbers of these requests in any order. Else output "Ok".

Limits

$1 \leqslant Q \leqslant 10^6$

$1 + r_k \leqslant i_k \leqslant 1000 - r_k$

$1 + r_k \leqslant j_k \leqslant 1000 - r_k$

$0 \leqslant r_k < 500$

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>6
10 10 5
10 10 4
5 10 0
10 5 0
10 15 0
15 10 0</pre></td><td><pre>Ok</pre></td></tr>
<tr><td><pre>2
6 6 5
11 11 5</pre></td><td><pre>2 1</pre></td></tr>
</table>

Problem G. Grouping (Division 1 Only!)

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

You are given N integers $a = \{a_i\}$. Let $b = \{b_i\}$ be such K numbers (not necessarily integers), that:

$$S = \sum_{i=1}^{N} \min_{1 \leqslant j \leqslant K} |a_i - b_j| \qquad minimal.$$

Find S.

Input

First line contains two numbers N and K. Second line contains exactly N integers — $\{a_i\}$.

Output

Only real number S with absolute or relative error not greater than 10^{-8}.

Limits

$1 \leqslant N \leqslant 5000$

$1 \leqslant K \leqslant N$

$0 \leqslant a_i \leqslant 400000$

Example

standard input	standard output
5 3 1 5 7 10 14	5

Remark for example:

It makes sense to take $\{1, 7, 14\}$ as $\{b_i\}$.

Problem H. Hidden triangles (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 mebibytes

N triangles are put on the plane in order from 1 to N. Whole inside region of each triangle is opaque and closes everything behind it.

Determine, which of the triangles are visible on the plane. I.e. have positive area, not covered above with any other triangle.

Input

In the first line number N is amount of triangles. Then in N lines triangles are listed in the order they were put to the plane. Each triangle is defined with 6 integers $x_{i1}, y_{i1}, x_{i2}, y_{i2}, x_{i3}, y_{i3}$, which are coordinates of each verticle. All the triangles are not degenerated. Any edge of a triangle has no more than one common point with any edge of anoter triangle.

Output

Output amount of visilbe triangles in the first line. List their numbers in any order in the second line.

Limits

$1 \leqslant N \leqslant 500$

$-1000 \leqslant x_{ij}, y_{ij} \leqslant 1000$, for $1 \leqslant i \leqslant N, 1 \leqslant j \leqslant 3$

Example

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

Problem I. Interactive

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

http://codeforces.ru/blog/entry/4037

Alex_KPR: — ...beginning somewhere from treaps, FFT and so on, coding and debuggig become straining and boring tasks...

Kunyavsky: — May I nitpick? Why are writing of Decarte's tree and FFT in same timescale?

Alex_KPR: — That's the first thing I remembered with high requirements of time and big space for mistakes.

Kunyavsky: — I've said: I'm nitpicking. On my opinion, here are two things, one of which is incredibly easier to code.

homo_sapiens: — Absolutely agree. FFT is written two or even three times faster than treap (but it's usually pushed with hands and feet)

Kunyavsky: — I meant quite the opposite. At the end, very specific topic that is.

homo_sapiens: — Seems really a matter of taste...

Time to find out, what's simpler.

Two sequences of N integers are offered: $\{x_i\}$ and $\{y_i\}$.

If you like FFT more, imagine these are coefficients of two polynomials:

$$P(z) = \sum_{i=0}^{N-1} x_i z^i$$

$$Q(z) = \sum_{i=0}^{N-1} y_i z^i$$

Find coefficients of polynomial $P(z) \cdot Q(z)$.

Though, if you like treaps more, imagine that $\{x_i\}$ are keys, and $\{y_i\}$ are priorities. Build treap by inserting into it all N elements (x_i, y_i) coherently. Output height of resulting treap after each insertion.

Those who chose FFT, recollect, that treap is binary search tree by keys, contained in (x_i). And in the same time it's a heap by priorities in nodes (y_i). I.e. for each vertex of the tree all the nodes from left subtree have lesser keys, than in top, and all vertices of right subtree have greater keys. And also priorities of sons are lesser than priority of top.

Independently of what you've chosen, author urges not to use prewritten code for this problem.

Input

The first line contains number N. The second has N different integers — $\{x_i\}$. The third line has N different integers, which make consequence $\{y_i\}$.

Output

If you chose FFT, display $2N - 1$ integers, which are coefficients of resulting polynomial. If treap, then N integers are heights of tree after insertion of each vertex.

Actually, you can display both, but then both results will be checked for correctness.

Limits

$1 \leqslant N \leqslant 50000$

$1 \leqslant x_i \leqslant 50000$

$1 \leqslant y_i \leqslant 50000$

guaranteed, that height won't be greater than 50

Example

standard input	standard output
6 4 1 2 3 6 5 1 6 3 4 5 2	4 25 20 34 54 71 72 58 56 37 10
6 4 1 2 3 6 5 1 6 3 4 5 2	0 1 2 2 3 4

Problem J. Joinery

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

You are given parallelepiped with length A, width B and height C. Find surface distance to the most distant (by surface) point from vertex of parallelepiped.

Input

The only line has three integers A, B, C.

Output

The only real number is distance with absolute or relative error no greater than 10^{-8}.

Limits

$1 \leqslant A, B, C \leqslant 1000$

Example

standard input	standard output
`1 1 1`	`2.2360679774998`

Problem K. K-edges graph (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second (*1.5 seconds for Java*)
Memory limit:	512 mebibytes

You are given an acyclic directed graph, consisting of N nodes and K edges. Find amount of edges in its transitive closure.

Transitive closure of graph G is a graph G', consisting of all nodes of orgignal graph G and such edges (u, v), that there is a path from node u to v in graph G.

Input

The first line contains two numbers N and K. Then K lines follow, each containing two integers a_i and b_i, descripting directed edge from a_i to b_i. Graph doesn't contain loops, cycles and multiple edges.

Output

Output only amount of edges in transitive closure.

Limits

$1 \leqslant N \leqslant 50000$

$0 \leqslant K \leqslant 50000$

$1 \leqslant a_i, b_i \leqslant N$

Example

standard input	standard output
5 6 1 2 2 3 3 5 4 5 1 5 1 3	7

Problem L. Lake (Division 2 Only!)

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

Alex is at the origin. But he wants to get to the shore of the lake, which is by some reason at the point (x, y). Probably, because tree grows there. At the shop nearby he bought speedwalking boots, with help of which he can step exactly by d distance.

Determine, what minimal number of steps is required to get to the destination. Point (x, y) doesn't match with origin.

Alex can step to any point of our flat world.

Input

The only line has three integers x, y, d.

Output

Output only minimal amount of steps.

Limits

$-10000 \leqslant x, y \leqslant 10000$

$1 \leqslant d \leqslant 10000$

Example

standard input	standard output
6 3 2	4

Problem M. Match them up (Division 2 Only!)

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

This time we'll have to search for maximal pair matching. And not simple, but lexicographically minimal.

You are given a bipartite graph of N vertices in the left set, N vertices in the right set and K edges between them.

Maximal pair matching is subset of edges of graph M with maximal power, such that any vertex of graph is incedent to no more than one edge from M.

Let $\{(u_i, v_i)\}$ set of edges be maximal pair matching, where u_i are vertices from left set and v_i are vertices from right set. Write all the vertices into one consequence in order: u_1, v_1, u_2, v_2, ..., u_m, v_m.

Find maximal pair matching with lexicographically minimal consequence of vertices.

Input

In the first line two numbers N and K are amounts of vertices in each set and amount of edges. Following K lines contain edges definition. Each edge is defined with a pair of numbers a_i and b_i, where a_i is a vertex from left set, and b_i is a vertex from right set.

Output

Output size of maximal pair matching m in the first line. Then write m lines with two numbers u_i and v_i each, where u_i is vertex from left set, and v_i is vertex from right set, corresponding to ith edge of the match.

Limits

$1 \leqslant N \leqslant 10^3$

$0 \leqslant K \leqslant 10^5$

$1 \leqslant a_i, b_i \leqslant N$

Example

standard input	standard output
3 5 1 2 1 3 3 2 2 3 2 1	3 1 3 2 1 3 2

Problem N. Need for sum thing (Division 2 Only!)

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

You are given a triangle table of N lines. First line consists of one element, second line of three, third of five and so on. ith line consists of $2i - 1$ elements. Central elements of all lines are in same coloum. So, lines form an isosceles triangle.

Each element of triangle is a number from '`0`' to '`9`'.

Problem is to respond queries with sums of elements of some triangle region of original table. Each request has a form like:

r_i, c_i, k_i, meaning that triangle of ith request has k_i lines, first line consists of one element (r_i, c_i). Second line of three elements $(r_i + 1, c_i - 1)$, $(r_i + 1, c_i)$, $(r_i + 1, c_i + 1)$, third of five and so on. Find sum of all the elements from the request.

Because of a great number of queries, they are generated programmly:

$r_1 = 1$

$c_1 = 1$

$k_1 = 1$

$r_i = r_{i-1} \cdot 10003 \mod N + 1$

$c_i = c_{i-1} \cdot 20003 \mod (2r - 1) + 1$

$k_i = k_{i-1} \cdot 30003 \mod (n - r + 1) + 1$

Input

In the first line N and Q are size of original triangle and amount of requests. Then table is defined in N lines. ith of them has exactly $(2i - 1)$ numbers, which are elements of table.

Output

Because of great number of requests, output sum of all the reports to requests.

Limits

$1 \leqslant N \leqslant 10^3$ $0 \leqslant Q \leqslant 5 \cdot 10^6$

Example

standard input	standard output
3 5 1 234 56789	67

Problem O. Open air (Division 2 Only!)

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

You are given a full set of dominoes with two numbers from 0 to N (N is even) written on each. There is only one exemplar of each pair of numbers $\{a, b\}$. It's easy to count that amount of dominoes is $\frac{(N+1)(N+2)}{2}$.

Place all dominoes horizontally in $N + 1$ row so, that sums of numbers on dominoes in each row are equal. Guaranteed, that it's always possible.

Input

Even number N in the only line.

Output

Output $N+1$ line with $N+2$ numbers in each: $a_1, b_1, a_2, b_2, \ldots, a_k, b_k$, $k = \frac{N+2}{2}$, where each pair a_i, b_i corresponds to a domino. Use each domino only once. Pair of numbers on each domino can be written in any order. If there are multiple solutions, you may output any one.

Limits

$2 \leqslant N \leqslant 100$, N is even

Example

standard input	standard output

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

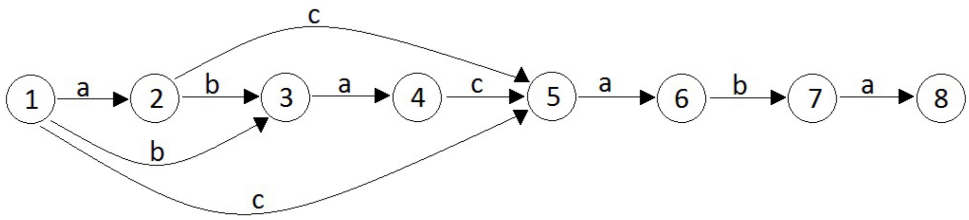
Problem P. Pseudo automaton (Division 2 Only!)

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

Given string S. Create determinstic finite-state automaton, that accepts all suffixes of S (and probably other finite strings). Automaton must consist of minimal number of states.

Each state of the automtaton is final. Starting state has number 1.

Image below shows automaton from test sample for string "`abacaba`"



Input

The only line contains string S, consisting of lowercase latin letters.

Output

In the first line, two numbers N and K are amount of states and transitions. t's followed by K lines, each having two numbers a_i, b_i and letter c_i, meaning transtion from state a_i to b_i by letter c_i.

You can output transitions in any order. If there are multiple solutions, output any of them.

Limits

$1 \leqslant |S| \leqslant 5\,000$

$1 \leqslant a_i, b_i \leqslant N$

'a' $\leqslant c_i \leqslant$ 'z'

Example

standard input	standard output

`abacaba`	`8 10`
	`1 2 a`
	`1 3 b`
	`1 5 c`
	`2 3 b`
	`2 5 c`
	`3 4 a`
	`4 5 c`
	`5 6 a`
	`6 7 b`
	`7 8 a`

Problem Q. Quiz (Division 2 Only!)

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

Almost everyone of you knows children's game ¡¡barley-break¿¿. At this problem you have to find a solution for some position.

The game includes square field $N \times N$, divided into cells 1×1. There are checks in each cells except of one. Number from 1 to $N^2 - 1$ is written on each cell. Each number occurs only once. At one turn you may move one of the chiecks next to the empty cell to it. Cell, where the check was, becomes empty. Checks cannot leave field.

Solving puzzle means placing checks in the specific order:

$$\begin{array}{ccccc} 1 & 2 & \dots & N-1 & N \\ N+1 & N+2 & \dots & 2N-1 & 2N \\ \vdots & & & & \vdots \\ N^2-N+1 & N^2-N+2 & \dots & N^2-1 & \end{array}$$

Empty place must be in the last cell of the last line.

For usual barley-break it looks so:

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & \end{array}$$

Position is given, find sequence of turns (not necessarily shortest, probably empty) which solves it. Or output, that position has no solution.

Input

The first line contains number N, that is size of the field. It is followed by N lines with N numbers in each. Numbers from 1 to $N^2 - 1$ correspond to checks, 0 corresponds to empty place. Each number from 0 to $N^2 - 1$ occurs only once.

Output

If position has no solution, write "`No`". Else write "`Yes`" in the first line and appropriate sequence of turns, written in second line without spaces. Turns are encoded by next way:

'`L`' means, that check to the left of the empty place must be moved to it.

'`R`' means, that check to the right of the empty place must be moved to it.

'`U`' means, that check above the empty place must be moved to it.

'`D`' means, that check below the empty place must be moved to it.

Turns number must not exceed 2 500 000. You may display any solutions if there are multiple ones.

Limits

$2 \leqslant N \leqslant 50$

$0 \leqslant a_{ij} \leqslant N^2 - 1$

Example

standard input	standard output
2 0 3 2 1	Yes DRULDR
2 2 1 3 0	No

Problem R. Reduction (Division 2 Only!)

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

You are given string S and list M, consisting of N words, each has length L. During an operation you may choose substring of string S, if it can be found as word in list M, and cut it out of string S. Then remaining parts of string S are merged if there are any. Determine minimal amount of opertions required to erase whole string S. It is guaranteed that it can be done.

Input

The frist line has word S. Second line contains integer N, that is amount of words in the list. It is followed by N lines listing words from M. All words consist of lowercase latin letters only.

Output

Output an integer, that is minimal number of opertions required to erase whole string S.

Limits

$1 \leqslant |S| \leqslant 100$

$1 \leqslant N \leqslant 100$

$1 \leqslant |M_i| \leqslant 100$

$1 \leqslant L \leqslant |S|$

Example

standard input	standard output
abacabada 4 aba aca ada abb	3