Problem A. Arithmetic Derivative

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

Lets define an *arithmetic derivative*:

- if $p = 1$ then $p' = 0$;
- if p is prime then $p' = 1$;
- if p is not prime then $n' = (a \cdot b)' = a' \cdot b + a \cdot b'$.

For example, $6' = (2 \cdot 3)' = 2' \cdot 3 + 2 \cdot 3' = 5$.

Given positive integers k and r, find all positive integers n such as $n \le r$ and $n' = k \cdot n$.

Input

Input contains two integers k and r $(1 \le k \le 30, 1 \le r \le 2 \cdot 10^{18})$.

Output

If there are no such numbers, print 0. Otherwise in first line print m — number of positive integers n does not exceeding r, for which $n' = k \cdot n$. Second line then must contain those m integers in ascending order.

Examples

standard input	standard input
1 100	2 4 27
1 2	0

Problem B. White Triangle

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

Given some triangle and two points A and B (they dont placed on the sides of triangle nor coincide with its vertices). For each point you know the distances to straight lines containing sides of triangle; additionally, you know if point lies inside or outside triangle.

Restore the coordinates of vertices of the triangle.

Input

First line of the input contains coordinates of point A. Second line contains distances between A and sides XY, YZ and ZX of triangle XYZ. If the point lies inside the triangle, all distances are positive, otherwise all distances are negative. Third and fourth line contains coordinates of point B and distances between it and same sides XY, YZ and ZX in similar format.

Coordinates are integers, does not exceeding 10^4 by absolute value, distances are non-zero integers, does not exceeding 10^4 by absolute value.

You may assume that triangle XYZ exists and coordinates of its vertices does not exceed 10^9 by absolute value.

Output

Print coordinates of vertices X, Y and Z, each vertice on the new line. Answer is considered correct, if for points A and B distances from sides of the printed triangle differ from the given ones by 10^{-5} or less.

Examples

standard input	standard input
1 5 1 5 5 5 2 5 5 2	0.00000 0.00000 0.00000 12.0000 16.0000 0.00000
5 5 5 5 5 5 -5 -5 -5 -13	20.0000 0.00000 0.00000 0.00000 0.00000 15.0000

Problem C. New Street

Input file:	`standard input`
Output file:	`standard input`
Time limit:	5 seconds
Memory limit:	256 mebibytes

The development company plans to build n houses on the one side of the street. Houses will differ only by number of floors.

Total number of floor in all n houses and maximum and minimum number of the floors on the some intervals was defined. When engineers of the company come with a solution, investors changed their mind... and again... and again.

Then engineers asked you to write a program to process requests of change and print number of acceptable solutions after the change.

More formally, here are m requests of two types:

- given l and r, change maximal and minial allowed number of floors in the houses from l-th to r-th, inclusively;
- change sum of floors of all n houses.

You must print number of different solutions after each request. Two solutions are considered to be different, if atleast one of houses at the same place have different number of floors.

Input

First line of the input contains three integers n, A and m ($1 \le n \le 10^9, 1 \le A \le 10^3, 1 \le m \le 10^5$) — number of places for the houses, initial sum of all floors and number of requests, respectively.

Each of the next m lines contain one request. Request description starts with integer t — type of the request.

If t is equal to 1, then it is request to change limitations on the number of flooors, and then 4 integers l, r, C, D ($1 \le l \le r \le n, 0 \le C \le D \le 10^3$) follow — in the houses from l-th to r-th, inclusively its allowed to build the houses with not less than C and not more than D floors.

If t is equal to 2, then request contain one integer A ($0 \le A \le 10^3$) — new total number of floors.

You may assume that number of requests of first type does not exceed 30.

Output

After each request print number of different solutions modulo 1000210433.

Examples

standard input	standard input
2 2 3 1 2 2 2 3 2 3 1 2 2 0 3	1 2 4
3 4 7 1 1 1 2 10 2 0 2 5 2 7 1 1 2 3 3 2 5 2 6	6 0 10 21 1 0 1
1000000 9 6 2 0 2 1 2 1000 1 1 100 10 10 2 1 2 1000	1 1000000 860010478 1 0 1

Problem D. Clones and Treasures

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

The magical treasury consists of n sequential rooms. Due to construction of treasury its impossible to go from room with biggest number to room in smallest number, i.e. from first hall one can go only to second, from second — to third etc.

Each room in treasury may be inhabitated by a clone of ancient king Koschey the Immortal, or, if it is not inhabitated by a clone, room may contain one gold coin or the invisibility potion which allows become invisible for time enought to pass only one room with a clone; each not inhabitated room contains exactly one of those items.

Ivan plans to enter the treasury and collect as much money as possible without being noticed by the clones of Koschey. He may choose the room where he starts his adventure, then go through some rooms; at any room he may cast the spell and leave the treasury, but then he will not be able to return anymore.

How much gold coins collects Ivan if he will act optimally?

Input

Input contains one string of length n ($1 \le n \le 10^6$) — map of the treasury. Rooms are listed from lest to right; character 'H' denotes invisibility potion, character 'M' — one gold coin, character 'K' — clone of Koschey. String cannot contain characters different than those ones.

Output

Print one integer — maximum amount of gold coins Ivan can collect.

Example

standard input	standard output
MKHMKMKHMHKKMHKMHHKKKHKMM	3

Problem E. Space Tourists

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

For the tourists outside the Solar system K^2 stations are built.

Tickets for the interstellar travels have unique ID — an integer, containing N digits in the base K. When tourist arrived to the solar system, he receives an two-digit number of station to arrive, which is subsequence if his ticket ID (i.e. may be obtained from it by removal $N - 2$ digits).

Now stations need some upgrade. Technical department wants to select some stations such as any tourist may be sent to one of selected stations, such as number of those stations will be as small as possible. Then other stations will be closed for the upgrade.

Given N and K, calculate minimum number of working stations, still able to service tourist with any ticket ID.

Input

Input contains two positive integers N and K ($2 \leq N \leq 100$, $2 \leq K \leq 100$).

Output

Print one integer — minumum number of selected stations.

Examples

standard input	standard input
4 10	34
12 10	10

Problem F. Matrix Game

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

Alice and Bob are playing the next game. Both have same matrix $N \times M$ filled with digits from 0 to 9.

Alice cuts the matrix vertically, choose order of the columns, then links all the columns to each other to have the cyclic sequence of $N \times M$ digits. Note that she cannot rotate the columns, i.e. end of some column must be linked to beginning of the other column. Then she cuts a sequence and reads the decimal representation of integer A upside down.

Bob cuts the matrix horizontally, choose order of the rows, then link all the rows to each other to have the cyclic sequence of $N \times M$ digits. Note that he cannot rotate the rows, i.e. end of some row must be linked to beginning of the other row. Then he cuts a sequence and reads the decimal representation of integer B from left to right.

Player who obtained the biggest integer wins. If both integers are are equal, then game is tied. You are given the matrix, find the number obtained by winner (or by both players in case of tie), if both Alice and Bob are playing optimally.

Input

First line contains integers N and M ($1 \le M, N \le 100$).

Each of next N lines contains string of M digits (without the spaces or other delimiters inside) — the given matrix. You may assume that atleast one digit in the matrix is not equal to 0.

Output

Print answer to the problem. Note that number must be printed without leading zeroes.

Example

standard input	standard input
2 2 28 27	8722

Problem G. Milkland

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

In the Milkland the Milk River is placed. This river can be represented as axis starting from some point (called the source of the river) in direction given by an integer vector. It is known that none of the Milkland cities is staying on the river (including the river source).

Given the coordinates of the Milkland cities, you need to connect them all with the roads such as:

- each road is segment of the straight line, connecting two cities;
- two roads cannot have common points (except for cases they are connecting the same site to two others);
- road cannot cross the river, but can touch river source;
- it can be possible to start at some city, visit all cities exactly once, and return to the same city.

Input

First line of the input contains one positive integer N, does not exceeding 10000. Second line contains four integers X, Y, dX, dY — coordinates of the river source and vector of river direction.

Each of next N lines describe one city and contain two integers X_i and Y_i. All integers in the input does not exceed 10^4 by absolute value.

You may assume that exists at least one pair of cities which can be connected by the road directly and at least one pair of cities which cannot be connected by the road because the segment intersects the river.

Output

Print N integers — numbers of cities in order they are connected in the path (each city is connected with two neighboring cities in this list, first city is connected with last one) or -1 if it is impossible to build the roads as described in problem statements.

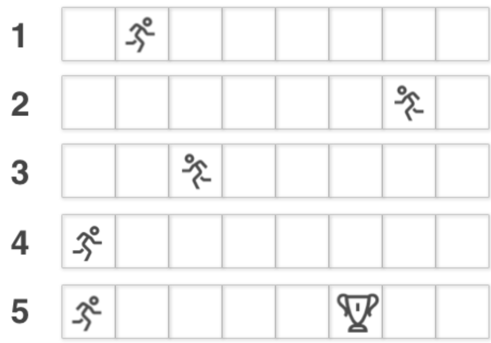
Example

standard input	standard input
20 0 0 0 -1 -2 3 2 3 4 0 -4 0 -3 -2 3 -2 3 -4 -6 -2 -6 2 6 2 -1 1 -5 -5 -4 -5 2 0 -3 -4 4 -5 1 1 -2 0 6 -2 5 -5	13 12 8 9 1 2 10 19 20 16 7 6 3 14 17 11 18 4 5 15 13

Problem H. Parallel Relay

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

The arena for the *parallel relay*, newly invented athletic discipline, consists of N horizontal tracks, each track of length M:



Before start of competition, for each athlete starting cell is chosen; additionally, in N-th row is chosen finish cell.

Each athlete runs on the his own track, the i-th athlete runs on i-th row. First athlete starts from her starting point, runs to the column containing starting cell of the second athlete and relays to her; then second athlete starts to run from his starting cell to the column, containing starting cell of third athlete and relays to her and so on; last athlete runs to the finish cell.

You are setting up the arena for coming competition. The federation sends to you dimensions of the arena, coordinates of starting cells in some rows (including row 1) and coordinates of finish cell. Lets call those rows Type 1 rows.

For remaining rows (lets define them as Type 2 rows) they send the distance between starting cell and end of track, but you are free to choose, it is distance from left side or from right side. Additionally, if several Type 2 rows are adjacent to each other, you are free to reorder those rows as you wish.

Competition will be translated on TV, and broadcasting company wants it to be as short as possible, so you want to do the choice such as summary distance for all athletes to run will be minimized.

To calculate time of the broadcasting, you want to calculate this minimum distance.

Input

First line of the input contains positive integers N and M — number of rows and columns at the arena ($1 \leq N, M \leq 10^4$).

Second line contains two integers between 1 and M — number of columns for starting cell for the first athlete and the finish cell for the last one.

Then $N - 1$ lines follow, i-th of it describes the i-th row in form "1 X" or "2 X": if the first number of line is equal to 1, then starting cell is in X-th cell from the left side; if its equal to 2, you may choose X-th cell from the left side or X-th cell from the right side.

Output

Print minimum summary distance for all athletes to run.

Examples

standard input	standard output
5 8 2 6 1 7 2 3 1 1 2 1	16
5 21 4 15 2 5 2 6 1 2 2 2	19

Problem I. Minimum Prefix

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

Consider string function $P(S, k)$ of string S and integer k.

$$P(S,k) = \begin{cases} S & \text{if } |S| \le k \\ S[1...k] & \text{otherwise} \end{cases}$$

Lets call $P(S, k)$ *k-prefix* of string S. Given N pairwise distinct strings. Find minimum k with next property: you can select some strings and shift them cyclically arbitrary number of times such as k-prefixes of all resulting strings will be pairwise distinct.

Input

First line of the input contains one integer N ($2 \le N \le 200$) — number of strings. Then N lines follow, each containing a string, consisting of lowercase Latin letters. Sum of lengths of all given strings does not exceed $2 \cdot 10^5$.

Output

Print minumum k with property mentioned in the problem statement.

Examples

standard input	standard output
4 aaaaa aabac abaaa ababa	2
5 noon oon noo no onn	3

Problem J. Terminal

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

N programmers from M teams are waiting at the terminal of airport. There are two shuttles at the exit of terminal, each shuttle can carry no more than K passengers.

Now employees of the airport service need to choose one of the shuttles for each programmer. Note that:

- programmers already formed a queue before assignment to the shuttles;
- each second next programmer in the queue goes to the shuttle he or she is assigned for;
- when all programmers, assigned to the shuttle, are in, shuttle immediately closes door and leaves the terminal;
- no two programmers from the same team may be assigned to the different shuttles;
- each programmer must be assigned to one of shuttles.

Check if its possible to find such as assignment; if the answer is positive, find minimum sum of *waiting times* for each programmer. Waiting time for a person is defined as time when shuttle with this person left the terminal; it takes one second to programmer to leave the queue and enter the assigned shuttle. At moment 0 the first programmer begins moving to his shuttle.

Input

First line of the input contains three positive integers N, M and K ($M \le 2000$, $M \le N \le 10^5$, $K \le 10^5$). Second line contains description of the queue — N space-separated integers A_i — ids of team of each programmer in order they are placed in the queue ($1 \le A_i \le M$).

Output

If it is impossible to assign programmers to she shuttles following the rules above, print -1. Otherwise print one integer — minimum sum of waiting times for all programmers.

Examples

standard input	standard input
7 3 5 2 2 1 1 1 3 1	39
12 3 9 1 1 1 2 3 2 2 2 2 2 2 2	116
2 1 2 1 1	4

Problem K. New Tetris

Input file:	`standard input`
Output file:	`standard input`
Time limit:	3 seconds
Memory limit:	256 mebibytes

In new version of the Tetris game used the field of N columns and M rows. At the bottom of the playfield (below each column) N integers are written. As game pieces squares 1×1 are used, each square contains an integer.

Here is process of the game.

1. New piece appears at the upmost row at the beginning of second
2. After that player selects a horizontal direction where to move the current piece (one row left, one row right or dont move at all).
3. At the end of each second the piece moves by one cell in direction selected by player (if this cell is busy with another piece or if it is outside the playfield nothing happens), then it moves one cell down (if this cell is busy with another piece or if it is outside the playfield, then this piece will not move any more and the score is updated.
4. If pieces moved down at step 3, go to step 2, otherwise go to step 1.

.

Score is calculated as sum of upmost integers for all columns.

While playing this game, next strategy was used: for each piece try to maximize score at the moment, when this piece stops its movement down. If it can be done by several sequences of choices, then select the sequence with least number of horizontal movements, and between them (if uncertainity is still not resolved) choose the sequence, leading to leftmost for the current piece position.

K pieces will appear, you need to predict the player's score after each piece. It is guaranteed that all K squares can be placed inside the playfield following described strategy.

Input

First line of the input contains two integers N and M ($1 \le N, M \le 10^5$) — number of columns and rows in the playfield, respectively.

Next line contains N positive integers written on the bottom of the playfield ($1 \le N \le 10^9$).

Third line contains one integer K — number of the pieces ($1 \le K \le 10^5$).

i-th of next K lines contains two integers col_i and x_i, describing i-th piece ($1 \le col_i \le N$, $0 \le x_i \le 10^9$): col_i denotes column where appears i-th square and x_i denotes integer on this square, respectively.

Output

Print K integers — score after each piece stopped its movement.

Examples

standard input	standard input
6 4 5 2 3 4 5 5 4 4 7 6 3 6 3 6 5	29 29 29 30
6 3 0 0 0 0 0 0 6 4 1 4 1 4 1 4 1 4 1 4 1	1 2 3 4 5 5

Problem L. Canonical duel

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

In the game «Canonical duel» board $N \times M$ is used. Some of the cells of the board contain *turrets*. A turret is the unit with four cannons oriented to north, east, south and west. Each turret can shoot exactly once. When turret is hit by the cannon of another turret, its activated. When turret is activated, all four cannons shoot simultaneously, then self-destruction process causes the turret to disappear.

Given the board with some turrets. You may add exactly one turret on one of cells which does not contains a turret and activate this new turret. Your goal is to destroy maximum number of turrets.

Input

First line of the input contains two positive integers N and M, does not exceeding 2000 — size of the board.

Each of the next N lines contain exactly M chars: '`+`' denotes that cell is occupied by a turret, and '`.`' that cell is empty.

Output

In the first line print maximum number of *existing* turrets you may destroy, then in second line print two space-separated integers — row and column of place where turret can be set. If it is impossible to destroy ever one turret in such a way, print only one line containing a zero; if several solutions exist, print any of them.

Examples

standard input	standard output
3 4 ++.. +... ..++	5 2 4
4 5 ++... ..+.. ....+ ...++	5 4 1
3 3 +++ +++ +++	0