ProCo 2017
Novice Division
Round 1

Problem A. How far you'll go

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

Before you embark on this journey to help Moana, you should write some words of encouragement to yourself.

Input

None

Output

Print a single string "I'll go pretty far!" (without the quotes).

Problem B. Meet up

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

Moana and Maui are trying to find each other. However the ocean is basically infinitely big so this is not an easy task.

The ocean can be viewed as a coordinate plane. In order for Moana and Maui to find each other, they have to be at the same coordinate at the same time.

Every minute, they both simultaneously move one unit up, left, right, or down. More formally, from (x, y), the allowed moves are to $(x - 1, y)$, $(x, y - 1)$, $(x + 1, y)$, or $(x + 1, y - 1)$.

Given the starting locations of Moana and Maui, determine if they can ever find each other.

Input

The first and only contains four space separated integers, x_1, y_1, x_2, y_2 $(-10^8 \leq x_1, y_1, x_2, y_2 \leq 10^8)$, the starting locations for Moana and Maui respectively.

Output

Print "YES" if Moana and Maui can meet, and "NO" otherwise.

Examples

standard input	standard output
`1 1 2 2`	`YES`
`-1 1 0 -3`	`NO`

Note

In the first sample, Moana starts at (1,1) and Maui starts at (2,2). In the first minute, Moana can go up to (1,2) and Maui can go left to (1,2) and they would meet.

Problem C. Braids

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

Moana likes to braid her hair with beads. To make her braids look beautiful, she wants to avoid having multiple beads of the same color in a row.

She has beads of k ($1 \leq k \leq 1000$) different colors, numbered from 1 to k. There are a_i beads of color i ($1 \leq a_i \leq 10^3$). Moana would like to thread all of those beads onto a single braid. Depending on how she arranges the beads, there might be several beads of the same color next to each other. Let L be the maximum length of a contiguous substring of beads that are the same color. Moana realizes that she can choose the order of the beads carefully to minimize L. What is the smallest possible L over all possible arrangements of the beads on a single braid?

Input

The first line contains a single integer b ($1 \leq b \leq 100$), the total number of braids. The next b lines each represent the beads that Moana would like to use for a single braid. In each line, the first integer is k ($1 \leq k \leq 1000$), corresponding to the number of bead colors. Then, k integers $a_1, \ldots, a_k$ follow, where a_i ($1 \leq a_i \leq 10^3$) represents the number of beads of color i.

Output

Print b lines, each containing exactly one integer L which is the smallest possible L for the corresponding braid in the input.

Example

standard input	standard output
2 1 1 3 6 1 1	1 2

Note

For the first braid, Moana has only 1 bead, so of course the answer is 1.

For the second braid, Moana has beads of 3 different colors. She has 6 of color 1, 1 of color 2, and 1 of color 3. The best she can do is have a maximum of 2 beads of the same color in a row. She can either arrange the beads as 11211311 or 11311211 to achieve this.

Problem D. Traveling

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

Moana wants to travel from Motunui to Lalotai. To do this she has to cross a narrow channel filled with rocks. The channel is represented by a 3 by n grid. Each cell contains a single bit: 1 if there is a rock at that cell and 0 if there isn't.

Moana's raft can only move to vertically or horizontally adjacent cells. Given the locations of rocks, determine if she can cross the channel, that is, if she can start from some cell on the left and get to some cell on the right without hitting any rocks.

Input

The first line contains n ($2 \le n \le 2 \cdot 10^4$), the width of the river in cells. The next 3 lines contain n space separated integers (either 0 or 1), denoting whether or not there is a rock at the cell.

Output

Print "YES" if Moana can cross the channel safely. Otherwise print "NO" (without the quotes).

Examples

standard input	standard output
3 1 0 1 0 0 1 1 0 0	YES
2 0 1 0 1 1 1	NO

Note

In the first sample, Moana can start in the middle cell on the left and go right, down, and then finally to the right.

In the second sample, all the cells on the right side have rocks, so there is no way to end on any cell on the right.

Problem E. Blight

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

Moana has come to a grassy field. The field is an n-row m-column grid where $(0,0)$ refers to the top left cell, and $(n-1, m-1)$ the bottom right cell. Each cell is either blighted or not. Entries in the grid can be modified via a special spell called "cross flip": place the cross shape illustrated shown below exactly over 5 cells in the field, and the cells covered by the cross will flip, meaning the ones affected by the blight will be cured, but the healthy ones will become blighted.

```
   _
 _|_|_
|_|_|_|
  |_|
```

In other words, a cross flip centered at (r, c) inverts the "blightedness" in positions $(r-1, c)$, $(r, c-1)$, (r, c), $(r+1, c)$ and $(r, c+1)$ for some r, c where $0 < r < n-1$, $0 < c < m-1$.

Given the status of the cells in the field, can Moana cure all blighted cells in the field by using the spell? If so, find the minimum number of cross flips necessary to do so. If it can't be done, print -1.

Input

The first line contains two integers space-separated integers n and m $(3 \leq n, m \leq 1000)$.

Then, n lines follow. On the r-th line $(1 \leq r \leq n)$, there are m bits in succession, each 0 or 1. The c-th bit $(1 \leq c \leq m)$ represents the cell $(r-1, c-1)$ and is 1 if the cell is blighted and 0 if not.

Output

Print a single line containing a single integer, representing the minimum number of cross flips necessary to cure all the blighted cells, or -1 if that is impossible.

Examples

standard input	standard output
3 4 0110 1001 0110	2
3 4 0100 1111 0100	-1

Note

In the first test case, Moana can perform a cross flip at $(1, 1)$ and $(1, 2)$. There is no single cross flip that will heal all the blighted cells. So it takes a minimum of 2 cross flips.

In the second test case, there is no sequence of cross flips that will make the whole field healthy.

Problem F. Bounce

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

Moana is playing a game. There are n ($1 \leq n \leq 10^5$) mini-coconuts in a 1-dimensional world on a 1-dimensional table, which spans from $-L$ meters (left side) to $+L$ meters (right side) ($1 \leq L \leq 10^6$). The i-th mini-coconut ($1 \leq i \leq n$) is initially at position x_i meters ($-L \leq x_i \leq L$) and is moving at a constant velocity of v_i meters per second ($0 < |v_i| \leq 1000$). Note that a negative velocity means moving left and a positive velocity means moving right.

All mini-coconuts basically have zero width, so they can be considered as point particles. However, they still bounce off each other. All mini-coconuts have the same mass and all collisions are perfectly elastic. Thus, when two coconuts collide they swap velocities.

Since all mini-coconuts have nonzero velocity initially, all the mini-coconuts will eventually fall off the table. Once a mini-coconut falls off the table, it can never get back on the table. Find the earliest time when there are no mini-coconuts left on the table.

Input

The first line contains two integers, n and L, representing the number of mini-coconuts and the radius of the table. Then, n lines follow. The i-th line ($1 \leq i \leq n$) contains two integers, x_i and v_i, representing the position and speed of the i-th mini-coconut.

Output

Print one line containing a decimal number representing the earliest time when there are no mini-coconuts on the table. For your answer to be considered correct, the absolute difference between your answer and the correct answer must be less than 1e-4.

Example

standard input	standard output
2 2 -2 1 2 -3	4.0000000000

Note

In the sample, there are two mini-coconuts, coconut 1 at -2 traveling right 1 at meter per second and coconut at +2 traveling left at 3 meters per second. They meet after 1 second at -1 and bounce off of each other, so coconut 1 now travels left at a speed of 3 meters per second and coconut 2 travels right at a speed of 1 second. From there coconut 1 falls off first and coconut 2 takes 3 seconds to travel 3 meters to fall off the right side. So it takes $1 + 3 = 4$ seconds for all the coconuts to fall off.

Problem G. Vines

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

The only dense patch of forest on Motunui is covered by a thick tangle of thorny poisonous vines. Moana would like to trim and destroy as much of the vines as possible. However, she only has a pair of large clippers and nothing else. It would be dangerous to clip a section of vine without people to hold them up, because then the vines could fall on Moana.

Several villagers have agreed to help her. They have decided to stick to the following plan. Two villagers will hold the two ends of a loose thornless section of vine, and Moana will cut its only remaining connection to the rest of the vines. This enables vines to be cut safely with no danger to either the villagers or Moana.

More formally, the vines are represented by a connected graph with n vertices ($1 \leq n \leq 10^6$) and m edges ($0 \leq m \leq \min(n(n-1)/2, 10^6)$). Trimming a section of the vines corresponds to the following operation on graphs:

- Find two distinct vertices a, b such that there is exactly one path between a and b. Let S, the section of vines, be the set of nodes on the path between a and b. - There can be at most one vertex $c \in S$ that is connected to vertices not in S. That is, S can be connected to the rest of the vines, but there must not be any thorns on S. - If there is such a vertex c, then let $d = c$. Otherwise, you can choose d to be any vertex in S. This will be the point that Moana will cut. - Delete all vertices in S except for d. The section of vine S is then removed from the tangle of vines.

Moana and the villagers have all the time in the world. They can cut as many sections of the vine as they want, and in any order. Will they eventually be able to remove all of the vines until a single vertex is left?

Input

The first line contains two integers n, m, representing the number of vertices ($1 \leq n \leq 10^6$) and number of edges ($0 \leq m \leq \min(n(n-1)/2, 10^6)$) in the vine graph respectively.

The next m lines contain two integers u_i, v_i each ($1 \leq u_i, v_i \leq n$) representing a single edge. This means that vertex u_i is connected to v_i. You are given that for every edge, $u_i \neq v_i$. You are also guaranteed that the edges are distinct. That is, for every $i \neq j$, $(u_i, v_i) \neq (u_j, v_j)$ and $(u_i, v_i) \neq (v_j, u_j)$.

Output

Print a single line containing only the word "YES" if the vines can be cleared, otherwise print "NO" (without the quotes).

Examples

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

Note

In the first test case, Moana can trim the section from vertex 1 to 3, picking $d = 1$. This results in only vertex 1 left, so the vines have been cleared.

In the second test case, Moana can first trim the section from vertex 3 to 4, where the section consists of vertices $3, 2, 4$. Since 2 is also connected to 1 which is not in the section, d must be 2. Then, only vertices 1 and 2 remain, so Moana can again trim that section to end with only one vertex left.

In the third test case, there is no way to trim sections of vines that results in there being only one vertex left.

Problem H. Raft Covering

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Moana needs to cover her raft with banana leaves.

Her raft is represented by $2 \times n$ grid, and she has available a bunch of banana leaves of dimension 1×2. She wants to lay all banana leaves on the raft such that no two banana leaves overlap and no part of a banana leaf is hanging off the edge of the raft. She is allowed to place leaves horizontally or vertically. How many ways can she entirely cover her raft with banana leaves in this way?

Moana already knows the answer is the n-th Fibonacci Number $F(n)$, where F is defined recursively as: $F(0) = 1, F(1) = 1, F(i) = F(i-1) + F(i-2)$ for $i > 1$.

However, if it were that easy Moana wouldn't have come to you for help. Moana's raft is no ordinary raft. There are k locations on her raft where she cannot cover with banana leaves (because something else must go there). You are given the coordinates of these locations. So the real task is to count the number of ways she can cover all parts of her raft except the k locations in the manner described above.

To make the problems easier, you only need to output the answer modulo $10^9 + 7$. You can also assume that k is always even, and no two forbidden locations lie in the same column.

Input

The first line of input consists of two integers n ($1 \leq n \leq 10^6$) and k ($1 \leq k \leq 10^5$). k is even.

The following k lines each consists of two integers c ($1 \leq c \leq n$) and r (0 or 1), indicating a forbidden location at column c (1 - indexed), row r (0 - indexed). All c's in the input are pairwise distinct.

Output

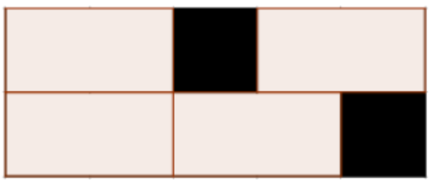
The number of different ways to cover the raft with banana leaves, mod $10^9 + 7$. Notice that this number can be 0.

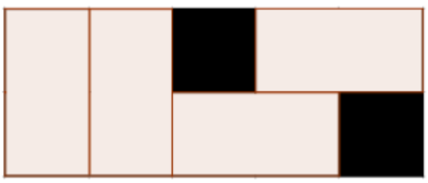
Examples

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

Note

In the first sample, there are two ways to cover the raft:





In the second sample, there are no ways to even place a leaf anywhere, and there are two empty spots, so there is no way to cover the raft.

Problem I. MIV

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

Moana has just invented a new text editor! She decided to call it Moana's Island Volcano (MIV) because she likes islands and volcanoes.

MIV operates on a single string. The string can contain only lowercase or uppercase alphabets a-z and A-Z, numbers 0-9, and underscores. Let the length of the text be L, where $(1 \le L \le 2 \cdot 10^3)$. There is also a cursor C that represents the current position in the text. The cursor is placed between and around characters in the text, so a cursor position of C means that the cursor is just before the character at index C in the text. The cursor is also allowed to be at the end of the text. You are guaranteed that the cursor must always be at a valid position, so $0 \le C \le L$ at all times.

MIV also knows what words are. Words are the maximal contiguous parts of the text separated by underscores. For example, the words in the text below are marked by asterisks underneath:

```
_Hello__world__Said123the______potato_3_times
 *****  *****  **********      ****** * *****
```

MIV allows you to perform the following operations that move the cursor:

- `h`: Moves the cursor left if it is not already in the beginning. The new cursor position is $\max(0, C-1)$.
- `l`: Moves the cursor right if it is not already at the end. The new cursor position is $\min(L, C+1)$.
- `w`: Moves the cursor to the next beginning of a word on the right of the cursor. The beginning of a word is the cursor position just before the first character of the word. If the cursor is currently at the beginning of a word, it still moves to the next beginning of a word on its right. If there are no more words to the right of the cursor, then the cursor moves to position L, the end of the text.
- `b`: Moves the cursor to the previous beginning of a word on the left of the cursor. If the cursor is currently at the beginning of a word, it still moves to the beginning of the previous word. If there are no more word beginnings to the left of the cursor, then the cursor moves to position 0, the very beginning of the text.

Apart from just moving the cursor around, MIV also has some operations that edit the text:

- `i [string]`: Insert the string (which contains only valid characters) at the current cursor position. After the string is inserted, the cursor also moves to the position right after the inserted string.
- `x`: Delete the character to the right of the cursor. That is, if the cursor is at position C, the character at index C is deleted, and the cursor stays at the same position. If the cursor is at the end of the text at position L, nothing happens.
- `d [movement]`: Delete the characters between the current cursor position and where the cursor would move if the movement is performed. Let $C_1 = C$ be the current cursor position, and let C_2 be the resulting position of the cursor if the movement is performed. Then assume $C_1 \le C_2$, otherwise swap C_1 and C_2. Delete the characters from index C_1 to C_2, including C_1 but not including C_2. If the cursor does not move, then no characters are deleted. After the characters are deleted, the new position of the cursor is $\min(C_1, C_2) = C_1$.

Sadly, MIV is just an idea at this point. Help Moana write MIV! Given an initial text S and a sequence of Q $(1 \le Q \le 1000)$ operations, print the resulting text after executing the operations. The cursor begins at position 0. You are guaranteed that the length of the text at any time (before and after all the operations) is at most 1000.

Input

The first line contains the initial text S, containing only valid characters. You are given that $0 \leq |S| \leq 1000$. The second line contains a single integer Q, representing the number of operations. The next Q lines contain 1 operation each. The operations are defined as above.

Output

Print a single line containing the resulting string. If the resulting string is empty, print an empty line.

Example

standard input	standard output
__hello_world_123__ 18 l d w w w b b d b h i __what__ x w w d w b b b d b i 123nice456	123nice456what__ello_world_

Problem J. Moana's Village

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

Moana's village consists of n huts labeled from 1 to n. Huts are connected by roads. It is guaranteed that from any hut it is possible to reach any other hut by following roads.

There are some special facts about Moana's village.

1. Each hut is a part of a *district* with exactly two other huts. A *district* is defined to be a group of three huts where for any two huts i, j within those three, there exists a road connecting hut i and hut j.

2. Each road belongs to exactly one district. More formally, for any road connecting huts i and j, there cannot exist two distinct districts that both contain huts i and j.

3. Each hut belongs to at most two districts.

Now, given the list of districts and the huts in them, Moana has q $(1 \le q \le 1000)$ questions for you. For each question, she wants to find the number of distinct simple paths from between two huts. A simple path is defined as any sequence of distinct huts where there is a road connecting every adjacent pair of huts in the sequence. Since this number can be large, output the answer modulo $10^9 + 7$. Moana guarantees that for each pair of huts that she asks a question about, both huts belong to exactly one district.

Input

To make the description of this structure easier, the input will be in the following format:

The first line consists of a single integer, k $(1 \le k \le 1000)$, denoting the number of districts in the village. It should not be hard to see that $n = 2k + 1$ always holds.

For the following k lines, each consists of three integers u, v, w $(1 \le u, v, w \le n)$, denoting the hut numbers in the corresponding district.

The next line consists of a single integer, q $(1 \le q \le 1000)$, denoting the number of queries.

For the following q lines, each consists of two integers a, b $(1 \le a, b \le n)$, denoting a query: number of simple paths between hut a and hut b. Remember that a belongs to exactly one district, and so does b (but a and b can belong to same or different districts). Output this answer modulo $10^9 + 7$.

Output

The output should consist of q lines. Each line consists of a single integer, denoting the answer of the corresponding query.

Example

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

Note

In the first query, there are four paths: $1 \to 3 \to 5$, $1 \to 2 \to 3 \to 5$, $1 \to 3 \to 4 \to 5$, and $1 \to 2 \to 3 \to 4 \to 5$.

In the second query, there are two paths: $1 \to 2$ and $1 \to 3 \to 2$.