

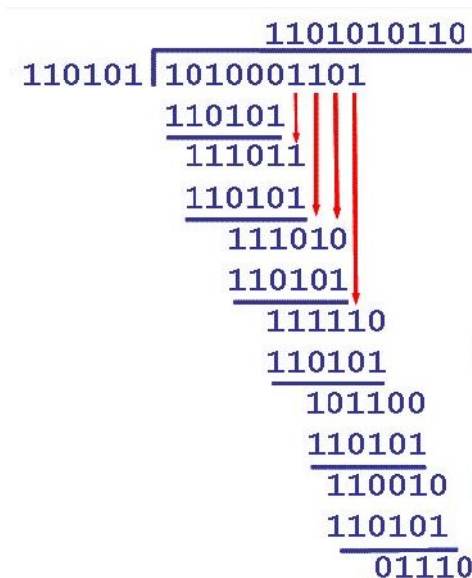
## Problem A. CRC Test

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

In the process of data transmission, bit errors (0 becomes 1, 1 becomes 0) may be caused by various interferences, which may lead to the receiver receiving the wrong data. CRC (Cyclic Redundancy Check) is a widely used algorithm to check whether there are errors in data transmission.

Specifically, CRC will select a predetermined  $k+1$  bits integer  $P$ , and add  $k$  redundancy bits as verification code after the  $n$ -bits data to be transmitted, so that the last transmitted  $n+k$  bits data can be divisible by  $P$  in modulo 2 division.

About modulo 2 division: the characteristic of modulo 2 operation is that it does not consider carry and borrow. Among them, module 2 addition is equivalent to module 2 subtraction and bitwise XOR; The modular 2 division method is equivalent to replacing all subtractions in vertical calculation with modulo 2 subtraction.



Now, given the divisor  $P$  used by cyclic check and the data transmitted, please calculate the verification code added in this transmission.

### Input

The first line of the input contains an integer  $T$  ( $1 \leq T \leq 1000$ ), denotes the number of testcases.

The first line of each test case contains a  $k+1$  ( $4 \leq k \leq 32$ ) bits binary number  $P$ , indicates the divisor used for cyclic verification.

The second line of each test case contains an  $n$ -bits ( $1 \leq n \leq 4000$ ) hexadecimal number  $x$ , indicates the data to be transmitted.

### Output

For each test case, output one line containing a binary number with a length of exactly  $k$ , representing the verification code of the data transmission you calculated.

## Example

standard input	standard output
3	01110
110101	11111101
28D	10000010
100000111	
1F1E33	
100000111	
ABCDEF123456	

## Problem B. Mr.X and Reviewing Location

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

One day, while playing video games before the final exam, Mr. X suddenly remembered the afternoon when he first went to the General Library of Wuhan University. At that time, X's dormitory suffered a power failure. He went to the library just to borrow a novel to read. When X entered the library hall, he found that the library was full. There were many people standing in the rotunda(a round hall) reviewing because they didn't get a seat. X was shocked and ashamed, so he borrowed the book and hurried away.

Since then, the grades of X who didn't work hard have been widened by the top graders, and he faced increasing pressure to fail. Having recovered his spirit, X could not help but feel some regret for not studying hard at the beginning. Suddenly a golden light flashed on the computer screen and X found himself traveling back in time to the afternoon of his first visit to the main library. This time he wanted to join the revising crowd in the hall, study well and work hard.

X found that, although there were already many people in the hall, people automatically kept a certain distance between each other, at least 2.0 meters away, to avoid mutual interference during the review. Given the location of the  $n$  people in the hall and the radius  $R$  of the rotunda (the center is at  $(0,0)$ ), can you help little X find a place to review in the rotunda, and be at least 2.0 meters away from the others?

### Input

The first line contains an integer  $n$  ( $1 \leq n \leq 10^5$ ), and a real number  $R$  ( $1.0 \leq R \leq 1000.0$ ), representing the number of people currently in the hall and the radius of the hall.

The next  $n$  lines, two real numbers per line, representing the coordinates of the position of a person already in the hall.

It is guaranteed that the current distance between any two people is at least 2.0 meters.

### Output

If the X cannot find the review location, print No, otherwise print Yes on the first line, and print two real numbers representing the coordinates of the reviewing location found for X on the next line.

When there is a valid reviewing location for Mr.X, your answer would be considered correct if and only if the location you give is inside or on the border of the rotunda, and is at least 2.0 meters away from the people currently in the hall, with the maximum absolute error of  $10^{-6}$ .

### Examples

standard input	standard output
6 2.000000005 -1.000000000 -1.732050808 1.000000000 -1.732050808 -2.000000000 0.000000000 2.000000000 0.000000000 -1.000000000 1.732050808 1.000000000 1.732050808	Yes -0.000000001 0.000000000
1 1.000000005 0.707106781 0.707100000	No

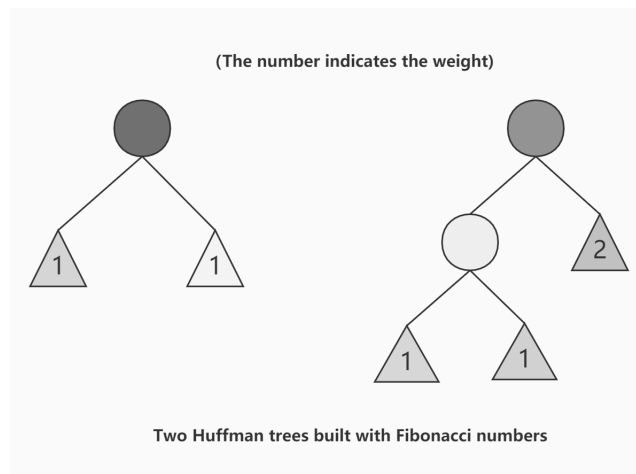
## Problem C. Data structure

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

At 8 o'clock in the morning, Little W and Little N are taking a data structure class. The teacher is talking about Huffman trees and Huffman coding.

In case you are not familiar with Huffman trees, here is a formal definition for it: for a given sequence  $(w_1, \dots, w_n)$ , the Huffman tree generated by  $(w_1, \dots, w_n)$  is defined as the rooted binary tree of  $n$  leaves which minimizes the value of  $\text{WPL} = \sum_{i=1}^n w_i d_i$ , where  $d_i$  denotes the depth of the  $i$ -th leaf of the tree. Recall that the leaves of a rooted tree are the non-root nodes that are connected to only one edge, and the depth of a node is defined as the number of edges of the simple path from the node to the root.

Little W is quite interesting in Fibonacci sequence recently, so he writes down the first  $K$  numbers  $F_1, F_2, \dots, F_K$  of the sequence. Recall that the Fibonacci sequence is a sequence satisfying  $F_i = F_{i-1} + F_{i-2}$  for any  $i \geq 3$ , and in this problem  $F_1 = F_2 = 1$ . He then draws an illustration of the Huffman tree generated by the  $K$  numbers  $(F_1, \dots, F_k)$ . Note that for a sequence of numbers, there may be many kinds of valid Huffman trees, and he will choose the one with the **lexicographically smallest** middle order (only leaf node). Little W is bored, so he paints Huffman the tree with **distinct colors on each node**. Let's name the tree  $\mathcal{W}$ . The following picture contains two examples.

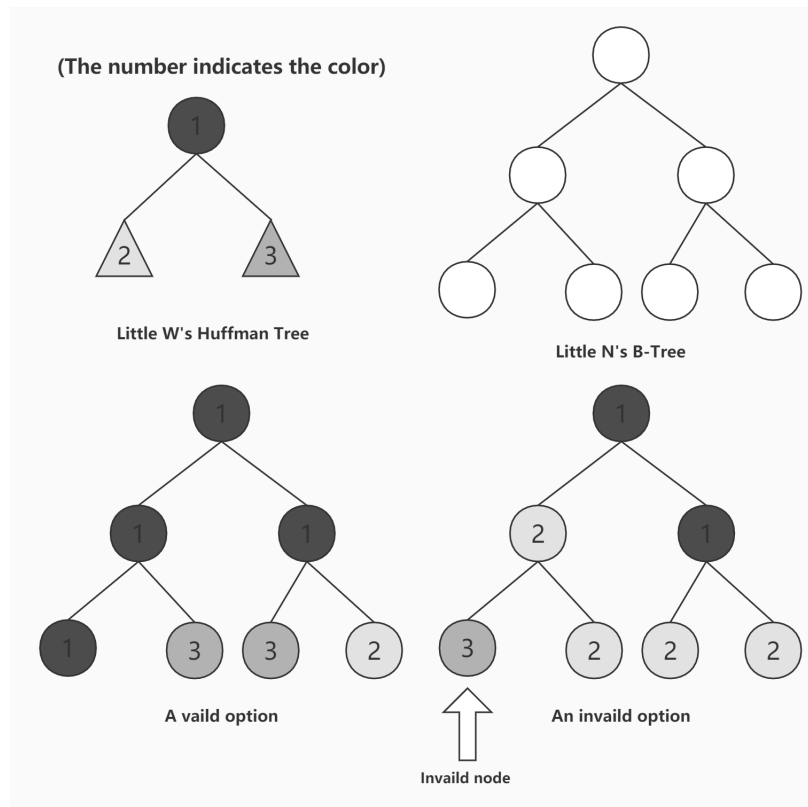


Little N has a crush on Little W recently and wants to attract his attention, so she wants to paint a tree similar to Little W's. She opens the textbook and finds a B-tree illustration and starts to paint it with color. This is a B-tree of  $M$  order and  $N$  nodes. You can simply consider a B-tree of  $M$  order as a tree in which each internal node has at least  $\lceil M/2 \rceil$  children and at most  $M$  children (an internal node is a node that is neither a leaf nor the root), while the root node can have less than  $\lceil M/2 \rceil$  children, and all leaf nodes have the same depth. Let's name the B-tree  $\mathcal{N}$ .

A painting option of  $\mathcal{N}$  is considered to be similar to  $\mathcal{W}$  if the following conditions are met:

- The colors used in  $\mathcal{N}$  appear in  $\mathcal{W}$ .
- The root of  $\mathcal{N}$  and the root of  $\mathcal{W}$  shares the same color.
- For any non-root node  $u$  in  $\mathcal{N}$  with its father node  $v$ , if they don't share the same color, then the color of  $u$  must be a son-color of the color of  $v$  in  $\mathcal{W}$ . Formally, let's define the color of  $u$  as  $A$ , the color of  $v$  as  $B$ , and  $\text{Node}_{\mathcal{W}}(X)$  as the node with color  $X$  in  $\mathcal{W}$ . Then the following must be held:  $A = B$ , or  $\text{Node}_{\mathcal{W}}(A)$  is the son of  $\text{Node}_{\mathcal{W}}(B)$ .

For example, in this picture below, the number indicates the color. The node which we figure out is invalid because color 3's father must be 3 or 1 (according to Little W's Huffman tree).



Can you help Little N find out how many painting options similar to  $\mathcal{W}$  there are? As the answer may be huge, you just need to print it modulo 998244353.

## Input

The first line contains a single integer  $T$  ( $1 \leq T \leq 10$ ) — the number of test cases.

The first line of each test case contains three integers  $K$  ( $1 \leq K \leq 1000000$ ),  $M$  ( $3 \leq M \leq 10$ ) and  $N$  ( $1 \leq N \leq 1000000$ ), indicating the length of Fibonacci sequence, the order of B-tree and the number of nodes in B-tree. Attention that we won't give you Little W's Huffman tree directly, and you need to build it yourself.

The second line of each test case contains  $N - 1$  integers, the  $i$ -th integer  $x_i$  ( $1 \leq x_i \leq N$ ) indicating that in Little N's B-tree the father of node  $i + 1$  is  $x_i$ , it's guaranteed that node 1 is the root.

It's guaranteed that the sum of  $N$  over test cases doesn't exceed 1000000 and the sum of  $K$  doesn't exceed 1000000.

## Output

For each test case, one integer in a single line indicating the answer (modulo 998244353).

## Example

standard input	standard output
2	9
2 3 3	361
1 1	
3 3 7	
1 1 2 2 3 3	

## Problem D. Fragmentation merging

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           **3 seconds**  
Memory limit:        **512 megabytes**

Yesterday is Monday, you have to attend the operating system class at 8:00 a.m. Sadly, you were so tired that you got up late and you didn't arrive at the class in time. So you decided to learn this class in your bed. In your dream, you have a memory space, which is a permutation of length  $n$ , where the  $i$ -th element is  $a_i$ .

In computer storage, fragmentation is a phenomenon in which storage space is used inefficiently, reducing capacity or performance and often both. The exact consequences of fragmentation depend on the specific system of storage allocation in use and the particular form of fragmentation. In many cases, fragmentation leads to storage space being wasted.

We call a pair  $(l, r)$  satisfying  $1 \leq l, r \leq n$  as a fragmentation. If  $l \leq r$ , this fragmentation contains elements  $a_l \dots a_r$  (including  $a_l$  and  $a_r$  themselves). At this time, this fragmentation can be regarded as a set contains  $r - l + 1$  elements. Otherwise, this fragmentation can be regarded as an empty set.

The most severe problem caused by fragmentation is causing a process or system to fail, due to premature resource exhaustion. In order to avoid this, the allocator may, instead of failing, trigger a defragmentation (or memory compaction cycle) or other resource reclamation, such as a major garbage collection cycle, in the hope that it will then be able to satisfy the request.

We define that if and only if any two fragmentations (which can be empty) meet the following conditions, then they can be merged. For the set  $A$  and  $B$  formed by any two fragmentations, let  $C = A \cup B$ , the largest element in  $C$  is  $c_{max}$ , and the smallest element in  $C$  is  $c_{min}$  (If  $C$  is an empty set, then  $c_{max} = c_{min} = 0$ ), the number of elements in  $C$  is exactly equal to  $c_{max} - c_{min} + 1$ , and  $A \cap B = \emptyset$ . We call the merged set  $C$  a super-fragmentation. Notice that a super-fragmentation is a set, not a pair.

For example, for memory space  $[1, 2, 4, 3, 5]$ ,

We define that fragmentation  $(1, 3)$  and fragmentation  $(4, 4)$  are mergeable, because set  $\{1, 2, 4\} \cup \text{set } \{3\}$  have 4 elements, which are exactly equal to  $4 - 1 + 1$ , and set  $\{1, 2, 4\} \cap \text{set } \{3\} = \emptyset$ .

Fragmentation  $(1, 2)$  and fragmentation  $(3, 3)$  can't be merged, because set  $\{1, 2\} \cup \text{set } \{4\}$  have 3 elements, not equal to  $4 - 1 + 1$ .

Fragmentation  $(1, 3)$  and fragmentation  $(2, 4)$  can't be merged, because set  $\{1, 2, 4\} \cap \text{set } \{2, 4, 3\} \neq \emptyset$ .

Now you want to know the number of super-fragmentations of different specifications can be obtained for arbitrarily selected two fragmentations. Formally, you need to calculate the number of sets  $C$ , which exists two fragmentations  $A, B$ , their combine result is  $C$ .

Notice that two sets  $A, B$  is the same if and only if  $A \subseteq B$  and  $B \subseteq A$ .

In order to simplify the problem, you only need to consider one merging, which means the super-fragmentation obtained after merging will not be merged with any fragmentation or super-fragmentation again.

### Input

There are multiple test cases. The first line of the input contains an integer  $T (1 \leq T \leq 10^4)$ , indicating the number of test cases.

For each test case, the first line contains an integer  $n (1 \leq n \leq 5000)$ , indicating the length of the permutation.

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$ , indicating the permutation in order.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^4$ .

## Output

For each test case, output the number of different kinds of super-fragmentation you can get in one line.

## Example

standard input	standard output
2	0
1	15
1	
5	
1 2 4 3 5	

## Problem E. Revue

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

Revue are enchanting stage performances with songs and dances. The path to Topstar will unfold for the stage girl who presents Revue's most brilliant show.

A total of  $n$  ( $1 \leq n \leq 10^6$ ) stage girls from Seisho Music Academy, with student numbers 1 through  $n$ , participated in the Talking Giraffe's auditions. Each stage girl has a radiance value  $r_i$  ( $i = 1, \dots, n$ ). They all have their own unique charm, so the radiance value is pairwise distinct. The auditions are made up of a number of Revue plays, and in each Revue, the winning girl gets more shine. But radiance doesn't come out of nowhere, it doesn't go out of nowhere, it just passes from one stage girl to another. In a Revue, if stage girl  $x$  beats  $y$ , this is the change in their radiance:

$$\begin{cases} r_x \leftarrow \max\{r_x, r_y\} \\ r_y \leftarrow \min\{r_x, r_y\} \end{cases}$$

Stage Girls' radiance is not easy to read, so you don't know the exact number of  $r_i$ . However, you get the script and know the next sequence of  $m$  ( $0 \leq m \leq 10^6$ ) Revues, with the winner  $x_i$  and loser  $y_i$  ( $i = 1, \dots, m$ ). You can continue to arrange  $w$  Revues in sequence at the end of the script, with the corresponding winners and losers. So there are total  $m + w$  Revues in the auditions.

But the stage the Talking Giraffe wants to show is the stage of fate that no one can predict. Depending on the choice of play, the stages can be quite diverse. The stage girls at Seisho Music Academy will be absent from  $k$  ( $0 \leq k \leq 10^6$ ) shows, which means up to  $k$  shows of the total  $m + w$  shows in Revue will NOT take place, and you don't know which shows they are.

Karen Aijo, whose student number is 1, did not want to fight with the partners in this baffling selection and become a lonely Topstar, she just wants to let everyone stars shine. If she doesn't get the maximum radiance value after the auditions, she won't be able to stand on the *Starlight* stage with Hikari-chan. So, she wants to know, after all the Revues are over, will she be able to get the biggest radiance value of all the stage girls?

Please write a program, help BaKaren, cleverly add  $w$  Revue at the end of the script and make add session  $w$  as little as possible, with the every possible initial value of  $r_i$  and Revue session that won't take place (at most  $k$ ), to let Karen Aijo eventually will surely be able to get the biggest radiance value, so that the all stars shine.

### Input

The first line contains three integers  $n$  ( $1 \leq n \leq 10^6$ ),  $m$  ( $0 \leq m \leq 10^6$ ),  $k$  ( $0 \leq k \leq 10^6$ ), which meanings are described in the statement above.

The next  $m$  lines contain two integers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n, x_i \neq y_i, i = 1, \dots, m$ ) in each line, representing the winner and the loser in each Revue the Talking Giraffe arranged.

### Output

Output an integer  $w$  ( $w \geq 0$ ) on the first line, representing the minimum number of session  $w$  to be added at the end of the script.

If  $w \leq 1000$ , then output  $w$  lines, each line contains two integers  $x, y$  ( $1 \leq x, y \leq n, x \neq y$ ), representing winners and losers of each Revue you arranged. You can output ANY possible solution.



## Examples

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

## Note

For the first example case, the final sequence of Revues is  $[(2, 3), (4, 5), (4, 1), (1, 3), (4, 2), (1, 4)]$ , and it's easy to verify that the maximum radiance will always be passed to person 1.

But if the Revue (1, 4) did not take place, the maximum radiance might remain at person 2 or person 4. That is the problem you face at the second example case.

## Problem F. Battery

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

Mr. X is a blogger of pilipili. He usually earns money by uploading videos taken by UAV(Unmanned Aerial Vehicle) to pilipili. One day, Mr. X comes to a new scenery spot. This scenery spot has  $M$  shooting locations. It takes a UAV to shoot a continuous video with a length of  $b_i$  seconds at  $i^{\text{th}}$  shooting location. The UAV uses a battery at one time to provide power, and the battery cannot be replaced during the shooting. But Mr. X only carried  $N$  UAV batteries, each with different capacity. The  $i^{\text{th}}$  battery can support UAV shooting for  $a_i$  seconds.

In order to attract more viewers, Mr. X hopes to shoot videos at as many shooting locations as possible. How many shooting locations can Mr. X shoot a video at most, with the  $N$  UAV batteries he carries, if he takes the best strategy? (assuming that it does not take time for the UAV to take off, land and switch to another shooting location)

### Input

The first line contains two integers:  $N, M$  ( $1 \leq N \leq 10^5, 1 \leq M \leq 3 \times 10^5$ ), which represent the number of batteries and the number of viewfinder positions.

Next line contains  $N$  integers  $a_1 \dots a_n$  ( $1 \leq a_i \leq 2^{30}$ ), which represent the capacity of the  $i^{\text{th}}$  UAV battery.

The third line contains  $M$  integers  $b_1 \dots b_m$  ( $b_i = 2^j, 0 \leq j \leq 30$ ), which indicate the shooting duration of the  $i$  shooting locations.

### Output

Output one integer indicates the maximum number of shooting locations Mr. Xie can shoot a video, with the best strategy.

### Example

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

## Problem G. Crossword Puzzle

Input file:           standard input  
Output file:         standard output  
Time limit:          1.5 seconds  
Memory limit:       256 megabytes

The Riddle Man is a crossword enthusiast who has a very unique crossword solving technique, so he can always solve puzzles very quickly. Today, the Riddle Man is doing a crossword puzzle with some bugs, he wants you to help him check his process, otherwise he won't tell you the meaning of the sign-in question.

A crossword is a word puzzle that usually takes the form of a rectangular grid of white- and black-shaded squares. The game's goal is to fill the white squares with letters, forming words or phrases, by solving clues, which lead to the answers. In languages that are written left-to-right, the answer words and phrases are placed in the white-shaded grid from left to right ("Across") and from top to bottom ("Down"). The black-shaded squares are used to separate the words or phrases.

The clues are given outside the grid, divided into an Across list and a Down list; the first cell of each entry contains a number referenced by the clue lists. For example, the answer to a clue labeled "17 Down" is entered with the first letter in the cell numbered 17, proceeding down from there. The numbered cells are numbered in the upper left corner of the cell and are numbered consecutively.

The Riddle Man is very good at solving clues, so he always guessed the answers to each clue and then tried to fill it in the squares. However, today's crossword puzzle is a little more difficult for the Riddle Man, so he has multiple possible answers to some clue and isn't sure which one is correct. So, your task is to help the Riddle Man check whether the answers he gives can complete the crossword puzzle. Since the Riddle Man is lazy, he also wants your program to output the final result of the puzzle for him, if his candidate answers complete the puzzle correctly.

The Riddle Man's anagrams will be presented in the form of character graphs. Recall that, the map of a word puzzle is a square or a rectangular grid of white- and black-shaded squares. Each grid is represented by a character picture of  $5 \times 5$ , characters ' ' said its lateral border, and ' | ' said its longitudinal border. For Black-Shaded Squares, the  $3 \times 3$  area in the center is filled with character '\*'; while the white grid is filled with spaces. In particular, for a white grid with number label, the number label is always in the upper left corner of the center area. For an answer-filled cell, the letter of the answer should always be in the center of the cell, and its number label is hidden if it has one. In addition, adjacent squares will share characters on the boundary. Examples are as follows:

```
-----
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
-----
| *** |
| *** |
| *** |
-----
| 1  |   |
|   |   |
|   |   |
-----
|   |   |
| A  |   |
|   |   |
-----
```

### Input

The first line of input contains four integers,  $H, W, N$ , and  $M$  ( $1 \leq H, W \leq 500$ ,  $1 \leq N + M < 1000$ ), indicating that the riddle consists of  $H \times W$  squares with  $N$  horizontal clues and  $M$  vertical clues.

The next  $4H + 1$  lines, each contains  $4M + 1$  characters (not including newline characters), represent the crossword map given by the Riddle Man;

Next  $N$  lines, each line begins with two integers  $a_i, c_i$  ( $1 \leq a_i \leq N + M$ ,  $c_i \in \{1, 2\}$ ),  $a_i$  represents the label of the  $i^{\text{th}}$  horizontal clue,  $c_i$  is the number of candidate answer for this clue by the Riddle Man. Then there are  $c_i$  strings, containing only uppercase English letters, with each string representing a candidate answer. The length of a candidate answer is at least 2 and will not exceed 500.

Next  $M$  lines, each line begins with two integers,  $d_i, c_i$  ( $1 \leq d_i \leq N + M$ ,  $c_i \in \{1, 2\}$ ),  $d_i$  represents the label of the  $i^{\text{th}}$  vertical clue,  $c_i$  has the same meaning as before, and the following  $c_i$  strings have the same restriction.

## Output

If the candidate answers provided by the Riddle Man can not complete the puzzle correctly, print No. Otherwise print Yes, and then print  $4H + 1$  lines of  $4M + 1$  characters per line (not including newline characters), representing the answer to the puzzle.

It is guaranteed that there is at most one solution to the given crossword puzzle.

## Examples

standard input	standard output
<pre> 4 5 3 2 -----  *** *** 1   *** ***   *** ***   *** ***   *** ***   *** ***  -----  *** 2         ***           ***          -----  3     *** ***         *** ***         *** ***  -----  ***   *** 4     ***   ***     ***   ***    ----- 2 2 SOLD HOLD 3 2 WHU HIS 4 1 AC 1 2 YOUR YOU 2 2 SHE HER </pre>	<pre> Yes -----  *** ***   *** ***   *** *** Y *** ***   *** ***   *** ***  -----  ***           *** S O L D   ***          -----        *** ***   W H U *** ***         *** ***  -----  ***   ***     *** E *** A C   ***   ***    ----- </pre>
<pre> 3 7 2 2 -----  *** *** 2   *** *** *** 4   *** ***   *** *** ***   *** ***   *** *** ***  -----  1     *** 3             ***             ***      -----  *** ***   *** *** ***   *** ***   *** *** ***   *** ***   *** *** ***  ----- 1 2 DDC ABC 3 2 CEO WAS 2 2 ACE MVP 4 2 LOVE PVP </pre>	<pre> No </pre>

## Problem H. Infomation Transmission

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

There are  $N$  stations of SERN in Akihabara, numbered 1 through  $N$ . Now SERN want to transfer a set of messages about Rintaro's time machine from station 1 to other stations. But because each station has a different broadcasting power, the signal may need to be relayed several times to reach each station. Specifically, all the sites form a directed graph, where an edge from  $X$  to  $Y$  indicates that  $X$  can transmit a signal to site  $Y$ . In each transmission process, the signal will occur attenuation, resulting in a part of the information get lost. However, if a message sent by a station decays up to **three** times after it is sent and it is picked up by the station itself, then we believe that the loop through which the message travels is capable of self-correcting. Each path along the entire loop that has the ability to self-correct will not generate signal attenuation.

Now SERN wants to know the minimum number of decays that a message from station 1 has to go through before it reaches each of these stations.

### Input

The first line has two integers  $N, M$  ( $1 \leq N \leq 300, 0 \leq M \leq 10000$ ), representing the number of stations and transitive relationships.

In the following  $m$  lines, two numbers  $X$  and  $Y$  in each line, representing that station  $X$  can transmit information to station  $Y$ .

It's guaranteed that the graph doesn't contain any multi edges or self loops.

### Output

Output one line containing  $N$  numbers, the  $i^{\text{th}}$  number represents the times of attenuation the information needs to go through from station 1 to station  $i$ . If a site cannot receive the information from station 1, please output  $-1$  instead.

### Example

standard input	standard output
8 11 1 2 2 5 2 3 3 5 3 4 4 5 5 1 4 6 6 7 7 8 8 6	0 0 0 0 0 1 1 1

## Problem I. Sequence

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

Nakisa likes sequence. She is interested in sequence  $P$  which meets the following conditions:

- Lenth of  $P$  is  $N$  and the index is from 1 to  $N$ .
- $\forall P_i \in [1, N]$  .
- There exists some restrictions  $(index, value)$  , indicating that  $P_{index}$  cannot be  $value$  ( $1 \leq index, value \leq N$ ) .

Nakisa does not care sequence itself, he only wants to know how many ordered tuples  $(A, B, L, R)$  satisfies  $\forall i \in [A, B]$  ,  $P_i$  can be any number in  $[L, R]$  , ( $1 \leq A, B, L, R \leq N$  ,  $A \leq B$  and  $L \leq R$ ).

For example,  $N = 2$ , one restriction is  $P_2$  cannot be 1. There are 5 vaild tuples, and they are as the follows:

- (1, 1, 1, 1)
- (1, 1, 1, 2)
- (1, 1, 2, 2)
- (1, 2, 2, 2)
- (2, 2, 2, 2)

An invaild tuple is (2, 2, 1, 1) because  $P_2$  cannot be 1.

### Input

First line contains two numbers  $N$  ( $1 \leq N \leq 5000$ ) and  $M$  ( $0 \leq M \leq 1000000$ ). Indicating the lenth of the sequence and the number of restrictions.

In the following  $M$  lines, each line contains two number  $a$  and  $b$  ( $1 \leq a, b \leq N$ ). Indicating a restriction that  $P_a$  cannot be  $b$ .

### Output

One number indicating the answer.

### Examples

standard input	standard output
2 1 2 1	5
3 5 1 1 2 1 3 1 3 2 3 3	9

## Note

- Attention that there may be some same restrictions.
- You are supposed to use faster IO method like **scanf** in C/C++ due to the huge amount of input.

## Problem J. Similar Triangles

Input file:           standard input  
Output file:         standard output  
Time limit:          1.5 seconds  
Memory limit:       256 megabytes

You are playing a game on a grid.

Given is a triangle with integer vertex coordinates. Your task is to find out a triangle with integer vertex coordinates, similar with the given triangle and it has the smallest area.

### Input

The first line contains a single integer  $T$  ( $1 \leq T \leq 10^4$ ) — the number of test cases. Then  $T$  test cases follow.

Each test case has one line, contains six integers  $x_A, y_A, x_B, y_B, x_C, y_C$  — the coordinates of three vertices (A,B,C) on the triangle.

It's guaranteed that  $-10^9 \leq x_A, y_A, x_B, y_B, x_C, y_C \leq 10^9$ , and three vertices A,B,C would not in same line.

### Output

For each test case, print six integers:  $x'_A, y'_A, x'_B, y'_B, x'_C, y'_C$  — the coordinates of three vertices on the triangle constructed by yourself.

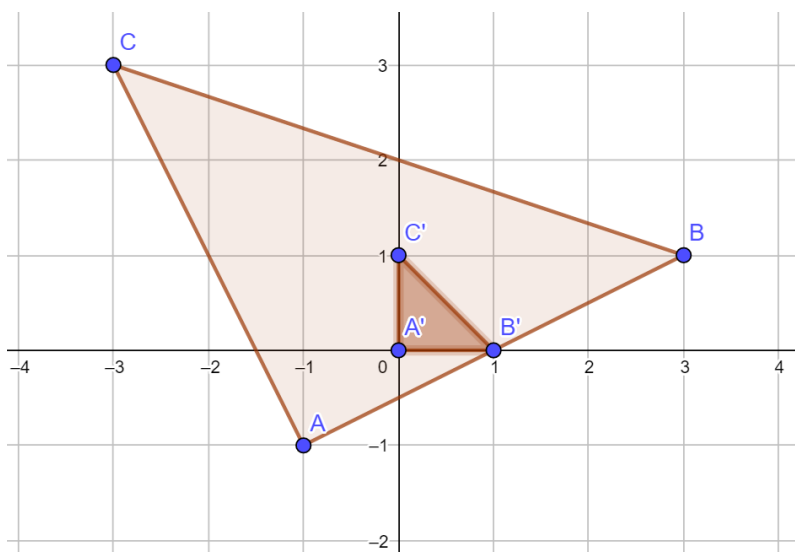
It's noticed that  $-10^9 \leq x'_A, y'_A, x'_B, y'_B, x'_C, y'_C \leq 10^9$  should be satisfied.

### Example

standard input	standard output
1 -1 -1 1 0 -2 1	0 0 -1 0 -1 -1

### Note

In the first test case, you can find out a triangle with integer vertex coordinates whose area is  $\frac{1}{2}$





## Problem K. Chtholly and World-End Battle

Input file:           standard input  
Output file:         standard output  
Time limit:          4 seconds  
Memory limit:       1024 megabytes

- *The life is not, for comparing things.*
- *So I hold both of light and shadow.*
- *The rays of stars, will bring me to you.*
- *Through the dark nights, I see the way to be shined.*

Chtholly gives you an integer sequence  $a$  of length  $n$ , and  $m$  queries. In each query, an interval  $[l, r]$  and a specific value  $v$  are given. You need to check every elements  $a_i$  in  $a_l, \dots, a_r$  successively, modify  $v$  to  $|v - a_i|$ , and tell Chtholly the value of  $v$  after the modification.

### Input

The first line contains two integers  $n, m$  ( $1 \leq n, m \leq 10^5$ ), representing the length of the sequence and the number of queries.

The second line contains  $n$  numbers, separated by spaces, representing the sequence  $a$ .

The next  $m$  lines, three integers  $l, r, v$  separated by spaces on each line, represents a query. The input numbers of the queries should take bitwise XOR to *lastans*, which equals to the answer to last query. For the first query, *lastans* equals to 0.

It is guaranteed that  $1 \leq a_i, v \leq 10^5$ , and  $l \leq r$  after decoding correctly.

### Output

For each query, output one line denotes the value of  $v$  you get after the modification.

### Example

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

### Note

The real value of  $v$  of the queries are 3, 1, 2, 1, 1, and the queried intervals are  $[3, 5]$ ,  $[2, 2]$ ,  $[1, 4]$ ,  $[1, 4]$ ,  $[3, 5]$ .