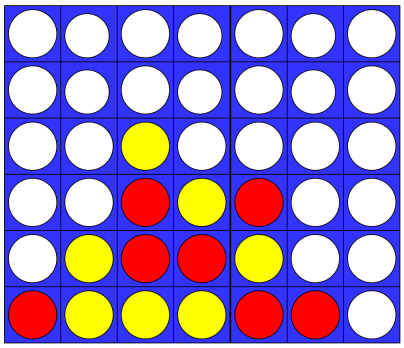
Connect 4
Problem ID: connect4

Connect 4 is a well known two player board game in which two players, Red and Yellow, take turns dropping discs of their respective colors into a 6×7 grid. Each new disc must be placed the lowest available spot in a given column. A player wins a game if they are the first to form a horizontal, vertical, or diagonal line of four discs with the same color.



We assume that Red always plays first. Given the sequence of Red and Yellow's moves, represented by a sequence of integers representing the column played, write a program to determine the winner of the game.

Input

Your program will receive input from standard input.

There will be 42 lines of input, each with a single integer between 1 and 7, representing the column chosen. The first line is Red's first move, the second line is Yellow's first move, the third line is Red's second move, etc.

Output

Your program should write to standard output.

Print exactly one line. If there is a winner, print the winner (either `RED` or `YELLOW`) and the winning turn, separated by a single space. Otherwise, print `DRAW`.

Scoring

There are 50 test cases, each worth 2 points. Your submission score will be the sum of the points you get from each test case you pass.

Sample Input 1	Sample Output 1
1 1 2 2 3 3 4 4 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7	RED 7

Treasure
Problem ID: treasure

There is a $n \times n$ grid and m treasures are located in the grid. You'd like to collect as many treasures as possible. You are initially at $(1, 1)$, but due to a broken leg, you can move only to the right or down, eventually reaching (n, n).

Write a program to determine the maximum number of treasures you can collect and the number of possible paths you could take to achieve this maximum treasure count based on the map of treasures.

Input

Your program will receive input from standard input.

The first line of the input contains an integer n. The following n lines each contain n characters indicating if there is treasure on that cell. The character will be 1 if there is treasure in the cell and 0 otherwise.

Output

Your program should write to standard output.

Print exactly one line containing the maximum number of treasures you can collect and the number of paths to achieve this maximum separated by a space. As the number of paths may be very large, please output the path count modulo $1,000,000,007$.

Constraints

- $2 \leq n \leq 10^3$
- $0 \leq m \leq n^2$

Subtasks

You will get points for each subtask when you pass all of the testcases of the subtask.

1. $n \leq 10$ (22 points)
2. $n \leq 300$ (31 points)
3. No additional constraints (47 points)

Sample Input 1	Sample Output 1
5 01010 00000 10010 10010 11000	4 3

Data Center
Problem ID: datacenter

There are n buildings in Quora city. The buildings are numbered from 1 to n. The location of each building can be represented as two-dimensional coordinates. Building i is located at (x_i, y_i).

We want to build a data center in one of the buildings. The data center will need to be connected to all other buildings in the city. The cost of a network connection between building i and building j is $\max(|x_i - x_j|, |y_i - y_j|)$. If we build a data center at building i, the total cost to set up network connections to all other buildings, denoted $D(i)$, is defined as the following:

$$D(i) = \sum_{j=1}^{n} \max(|x_i - x_j|, |y_i - y_j|)$$

Find the best building to have a data center such that we minimize the total cost of setting up network connections.

Input

Your program will receive input from standard input.

The first line contains a positive integer n representing the number of buildings. In the following n lines, the i-th line contains two positive integers x_i and y_i representing the location of building numbered i, starting with building 1.

Output

Your program should write to standard output.

Print exactly one line containing the number of the best building to construct the data center. If there are multiple possible answers, print the answer with the smallest building number.

Constraints

- $1 \leq n \leq 5 \cdot 10^5$
- $1 \leq x_i, y_i \leq 10^9$

Subtasks

You will get points for each subtask when you pass all of the testcases of the subtask.

1. $n \leq 10^3$ (9 points)
2. $x_i = y_i$ (20 points)
3. No additional constraints (71 points)

Sample Explanation

$$D(1) = \max(|1-3|, |4-1|) + \max(|1-5|, |3-4|) = 7$$
$$D(2) = \max(|3-1|, |1-4|) + \max(|3-5|, |1-3|) = 5$$
$$D(3) = \max(|5-1|, |3-4|) + \max(|5-3|, |3-1|) = 6$$

Building 2 is the best building since its cost is the lowest.

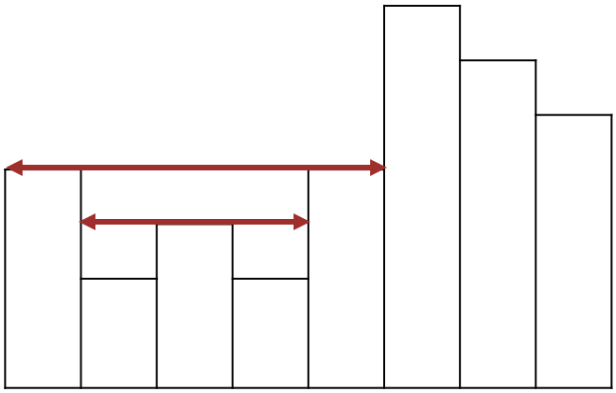
Sample Input 1	Sample Output 1
3 1 4 3 1 5 3	2

Skyscraper

Problem ID: skyscraper

There are n skyscrapers. They are arranged in a row from left to right and numbered from 1 to n. Skyscraper i has height h_i. We say that skyscraper i can see skyscraper j if there is no skyscraper higher than skyscraper i between the two skyscrapers and $h_j \leq h_i$.

For example, assuming that $n = 8$ and $h = [4, 2, 3, 2, 4, 7, 6, 5]$. For the skyscraper 3, it can see 3 skyscrapers from skyscraper 2 to skyscraper 4. For the skyscraper 5, it can see 5 skyscrapers from skyscraper 1 to skyscraper 5. This is represented in the figure below.



There's a lot of construction, and we'd like to keep track of which skyscrapers can see each other. In particular, we'd like to build a program supporting three types of queries:

- $\text{num\_visible\_skyscrapers}(i)$: returns the number of skyscrapers that skyscraper i can see.
- $\text{set\_height}(i, x)$: updates the height of skyscraper i to x. (i.e. $h_i = x$)
- $\text{set\_height\_range}(i, j, x)$: updates the heights of skyscrapers from i to j to x. (i.e. $h_i = h_{i+1} = \cdots = h_j = x$) where $i \leq j$.

You are given initial heights the skyscrapers and q queries. These q queries should be processed in an order. Write a program to handle these queries, and print any result from $\text{num\_visible\_skyscrapers}$.

Input

Your program will receive input from standard input.

The first line contains two space-separated positive integers n and q representing the number of skyscrapers and the number of queries. The next line contains n positive integers. The i-th integer represents the initial height of skyscraper i, h_i. In the following q lines, the i-th line contains several positive integers representing the i-th query. Each line is in one of the three following formats:

- `1 i`: represents $\text{num\_visible\_skyscrapers}(i)$
- `2 i x`: represents $\text{set\_height}(i, x)$
- `3 i j x`: represents $\text{set\_height\_range}(i, j, x)$

You may assume that there is at least one $\text{num\_visible\_skyscrapers}$ query.

Output

Your program should write to standard output.

Print b lines, one line for each of the b $\text{num\_visible\_skyscrapers}$ calls, each containing the result of that.

Constraints

- $1 \leq n, q \leq 3 \cdot 10^5$
- $1 \leq h_i \leq 10^9$
- $1 \leq i \leq n$ for num_visible_skyscrapers and set_height queries
- $1 \leq i \leq j \leq n; 1 \leq x \leq 10^9$ for set_height_range queries

Subtasks

You will get points for each subtask when you pass all of the testcases of the subtask.

1. $n, q \leq 2 \cdot 10^3$ (11 points)
2. Only num_visible_skyscrapers is given (23 points)
3. Only num_visible_skyscrapers and set_height are given (37 points)
4. No additional constraints (29 points)

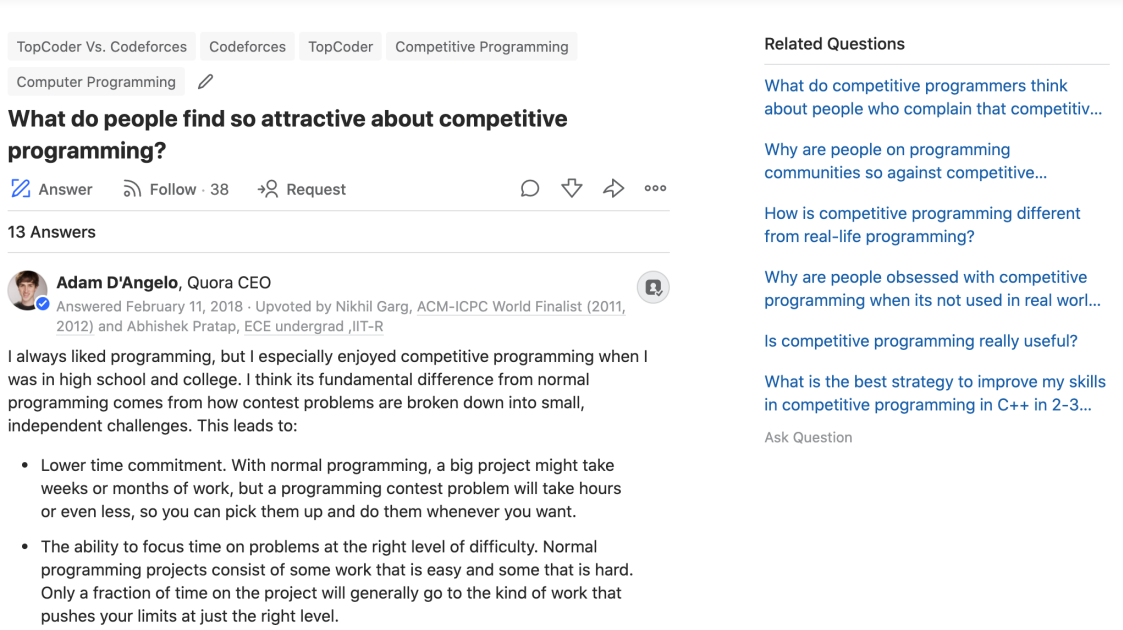
Sample Input 1	Sample Output 1
8 6 4 2 3 2 4 7 6 5 1 3 1 2 2 3 8 1 5 3 5 7 1 1 8	3 1 2 5

Help Crawlers

Problem ID: crawlers

The heart of Quora is questions. When a user lands on a question page, we provide related questions so that they can browse more questions. In order to distribute our content to a broader audience, we can leverage crawlers. When a crawler lands on a question page, it will visit all related questions on that page which it hasn't yet visited.

Given the current mapping from each question to its related questions, determine a minimum set of additional related questions which we could add such that a crawler could reach all other questions starting from any question on Quora.



Input

Your program will receive input from standard input.

The first line of the input contains two integers n and m, where n is the number of questions and m is the number of related question pairs. After that, there are m lines. The i-th line contains two positive integers $q_{i,1}$ and $q_{i,2}$ indicating that $q_{i,2}$ is a related question of $q_{i,1}$.

Output

Your program should write to standard output.

The first line should print the minimum number of related question pairs you need to add. After that, print the question pairs in the following lines. Each line should have two questions indicating that the second question is a related question for the first question.

Constraints

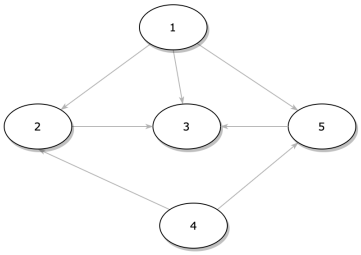
- $1 \leq n \leq 10^6$
- $0 \leq m \leq 2 \cdot 10^6$
- $1 \leq q_{i,1}, q_{i,2} \leq n; q_{i,1} \neq q_{i,2}$
- There is no self edge
- There may be multiple edges

Subtasks

You will get points for each subtask when you pass all of the testcases of the subtask.

1. $n \leq 10; m \leq 20$ (19 points)
2. $n \leq 10^3; m \leq 2 \cdot 10^3$ (32 points)
3. No additional constraints (49 points)

Sample Explanation



This image illustrates the sample Input/Output.

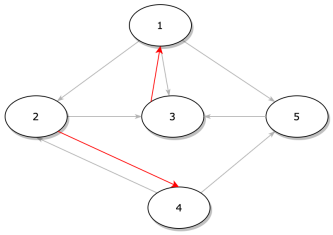
If a crawler lands on question 1, it can reach questions 2, 3, 5.

If a crawler lands on question 2, it can reach question 3.

If a crawler lands on question 3, it cannot reach any other question.

If a crawler lands on question 4, it can reach questions 2, 3, 5.

If a crawler lands on question 5, it can reach question 3.



With two more related question pairs, we can make crawlers reach all questions.

Sample Input 1	Sample Output 1
5 7 1 2 2 3 1 3 1 5 4 5 4 2 5 3	2 3 1 2 4

[ML] Broken Message

Problem ID: message

You were given a secret encoded message to deliver, but accidentally spilled water on the paper and lost part of every line in the message. You need to repair the message before anyone discovers your mistake.

Thankfully, you know that the encoding simply maps words to a number, and the original language was English. For example, the encoding map might be:

the	1
apple	2
down	3
fell	4
well	5

"`the apple fell down the well`" in this scheme would be encoded as `[1,2,4,3,1,5]`.

Unfortunately, you don't know the encoding map, but still need to find a way to write reasonable sentences. You only lost one word of each sentence, so you think you should be able fill in the missing blanks.

Write a program to determine the number representing the missing word on each line. Note that this task awards points for partial solutions as it's not possible to perfectly determine every missing word.

Input

Your program will receive input from standard input.

The first row will be a single integer n, representing the number of sentences in the message. n lines will follow, each of which contains a list of between L_{min} and L_{max} space separated integers each representing either a single word, w_i or -1 if this is the missing word. Each line will have exactly one missing word.

Output

Your program should write to standard output.

Print n lines, each with a single integer to replace the missing word.

Constraints

- $10^3 < n < 5 \cdot 10^3$
- $L_{\text{min}} = 4$
- $L_{\text{max}} = 12$
- $0 \leq w_i < 10^3$

Scoring

Points awarded for this problem is defined as

$$100 \times \frac{\min(\text{percent\_correct}, 30)}{30}$$

That is, correctly identifying 30% of the missing words awards full points, while solutions below this receive points proportionally. Scores will be rounded to the nearest integer value.

Sample Explanation

In this example, the first input is missing the 3rd word, and the 2nd input is missing its 6th word. The output suggests filling in the missing word in the first input with 6 and the missing word in the 2nd input with 4.

Sample Input 1	Sample Output 1
2 1 2 -1 5 4 1 3 2 6 5 -1	6 4

[ML] Malicious Data

Problem ID: malicious

You're a spy and as part of your mission, you need to sabotage a machine learning model to make it less accurate. You've gotten access to the training data, but only can add a small number of rows to it in order to avoid detection. The existing training data consists of k feature rows of d features, each with a corresponding binary label. You may add m additional feature/label pairs to the training data, with the goal of making the resulting model as inaccurate as possible.

You can download the script we'll be using to train the model in the attachments of this problem, as well as the training data. For this problem, you only need to submit the output as a `txt` file, not the code used to generate the output.

Input

The first line consists of three space separated integers, k, d, and m. The next k lines consist of d space separated floating point feature values. The final k lines consist of either 1 or 0, representing the labels.

Output

Your program should write to a `txt` file which you should then upload.

Write $2m$ lines. The first m lines should each contain d space separated floating point numbers representing the features for added data points. The next m lines should contain 1 or 0, representing the labels for the newly added data points.

Scoring

Scoring for this problem will be based on the change in (hidden) test data ROC-AUC of a gradient boosted decision tree trained on the training data before and after your additions to the training data. In particular, score is defined as:

$$100 \times \min(1, (\text{AUC}_{\text{before}} - \text{AUC}_{\text{after}}) \times 10)$$

so a 0.1 change in AUC will result in full points, where AUC is a float in [0,1] as returned by `sklearn.metrics.roc_auc_score`. Scores are rounded to the nearest integer.

Sample Input 1

```
10 5 3
0 3 2 0 7
5 9 0 2 7
2 9 2 3 3
2 3 4 1 2
9 1 4 6 8
2 3 0 0 6
0 6 3 3 8
8 8 2 3 2
0 8 8 3 8
2 8 4 3 0
1
1
1
1
0
1
0
1
0
0
```

Sample Output 1

```
3 0 0 2 0
1 2 1 1 1
1 1 1 1 1
0
0
1
```