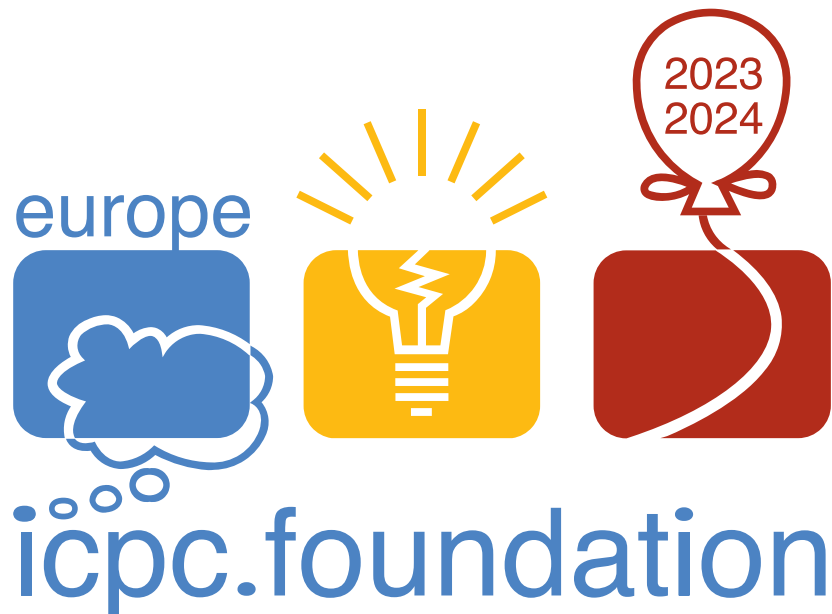


ICPC SouthWestern Europe Regional Contest 2023–2024

Paris, 28 January 2024

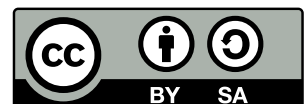


Judges and Problem Setters

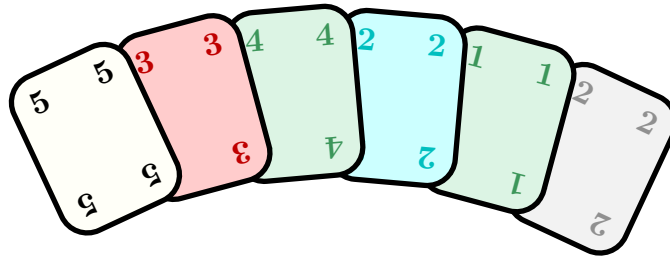
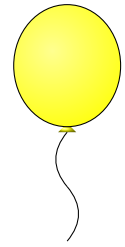
- Guillaume Aubian (Charles University)
- Mehdi Bouaziz (Atacama)
- Nofar Carmeli (Inria Montpellier)
- Thomas Deniau
- Garance Gourdel (GitGuardian)
- Ioana Ileana (Université Paris Cité)
- Vincent Jugé (Université Gustave Eiffel), *chief judge*
- Raphaël Marinier (Google)
- Amaury Pouly
- Pablo Rotondo (Université Gustave Eiffel)
- Philippe Schnoebelen (CNRS & ÉNS Paris-Saclay)
- Cheng Zhong (Google Zurich)

This problem set consists of 13 problems, on 29 pages.

This work is licensed under a Creative Commons “Attribution-ShareAlike 4.0 International” license.



A: Card game



The Olympic Games in Paris are almost here, and you are thrilled! You and four of your friends managed to get five of the rare tickets for sports climbing even though all tickets were sold out in less than two hours! While waiting for the games to start, you decide to do something fun to keep your mind off the wait: play your favourite card game.

The card deck contains four standard suits of different colours: silver (S), white (W), emerald (E), and red (R), as well as one trump suit coloured cyan (C). That is, the Cyan cards outrank all other cards. There are N cards of each suit, numbered from 1 to N . This means that, in total, the deck comprises $5N$ cards. At the beginning of the game, the deck is randomly distributed between the five players, such that each player gets N cards.

Before you start playing, you want to organise your cards such that all cards of the same suit are next to each other in increasing order, and the trump cards appear at the end (also in increasing order). When you receive your cards, they appear in your hand as a sequence. To organise them, you perform a sequence of actions, where in each action you take one card out of your hand and put it back in your hand at another position (between two cards, before the first card, or after the last card).

You cannot help but wonder: what is the minimum number of actions you need to take in order to organise your hand?

Input

The input consists of two lines. The first line contains the number N . The second line contains N space-separated values describing the sequence of cards in your hand. Each value is composed of one letter of the set $\{S, W, E, R, C\}$ (describing the card suit), followed by an integer V such that $1 \leq V \leq N$ (describing the card number).

Output

The output should contain a single line, consisting of a single number: the minimum number of actions required to organise your hand.

Limits

- $1 \leq N \leq 100\,000$.

Sample Input 1

```
4
C1 R2 E4 R1
```

Sample Output 1

2

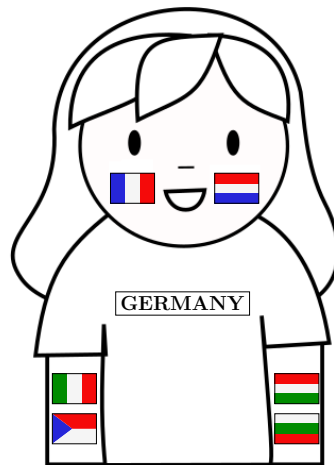
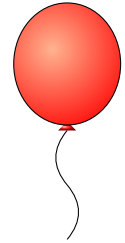
Sample Input 2

5
S2 W4 E1 R5 C1

Sample Output 2

0

B: Supporting everyone



Alice is attending a sport event with many national teams and one thing is important to her: supporting every country.

There are N countries represented and she has two ways to support a country: either have the flag drawn on her or have a pin with the name of the country. Alice has a list containing, for each country, the colours needed to make its flag. A total of M colours that may appear across all flags and, in Alice's list, each colour is conveniently represented as an integer between 1 and M .

Each crayon and pin cost 1, but her budget is tight... Can you help her find the minimum she can spend to support everyone?

Input

The first line contains the two space-separated numbers N and M . Then follow $2N$ lines, grouped in pairs; the $(2i - 1)^{\text{th}}$ and $2i^{\text{th}}$ lines represent the i^{th} country. More precisely, the $(2i - 1)^{\text{th}}$ line contains a single integer k_i : the number of colours in the flag of the i^{th} country. Then, the $2i^{\text{th}}$ line contains k_i space-separated numbers $c_{i,1}, c_{i,2}, \dots, c_{i,k_i}$; these are the colours in the flag of the i^{th} country.

Output

The output should contain a single line, consisting of a single number: the minimum amount Alice can spend on crayons and pins to represent every country.

Limits

- $1 \leq N \leq 1\,000$;
- $1 \leq M \leq 100$;
- $1 \leq k_i \leq M$ for all $i \leq N$;
- $1 \leq c_{i,j} \leq M$ for all $i \leq N$ and $j \leq k_i$;
- for all $i \leq N$, the M colour numbers $c_{i,j}$ are pairwise distinct.

Sample Input 1

```
7 6
3
1 4 5
3
1 4 5
3
1 4 5
3
3 4 5
3
3 4 5
3
3 4 5
3
2 5 6
```

Sample Output 1

```
5
```

Sample Explanation 1

The three first countries could be France, the Netherlands, and the Czech Republic, all represented by blue (1), white (4), and red (5). The three next countries could be Italy, Hungary, and Bulgaria, with green (3), white (4) and red (5). The last one could be Germany, with black (2), red (5), and yellow (6). The minimum cost is 5: we buy four (blue, green, white, and red) crayons and one pin (for Germany).

Sample Input 2

```
8 12
2
7 9
12
1 2 3 4 5 6 7 8 9 10 11 12
2
7 9
2
7 9
3
3 4 11
2
7 9
2
7 9
2
7 9
```

Sample Output 2

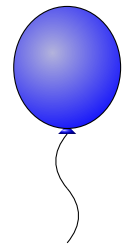
```
4
```

Sample Explanation 2

We can buy two crayons for the colours 7 and 11 and buy two pins for a total cost of 4. All six countries with flag colours 7 (red) and 11 (white) could be Canada, Indonesia, Japan, Malta, Monaco, and Poland. The flag of Belize has 12 colours, including red and white, and the fifth country could be Botswana.

Page intentionally left blank

C: Metro quiz



Two Olympics spectators are waiting in a queue. They each hold a copy of the metro map of Paris, and they devised a little game to kill time. First, player A thinks of a metro line (chosen uniformly at random among all metro lines) that player B will need to guess. In order to guess, player B repeatedly asks whether the line stops at a metro station of her choice, and player A answers truthfully. After enough questions, player B will typically know with certainty which metro line player A had in mind. Of course, player B wants to minimise the number of questions she needs to ask.

You are given the map of the N metro lines (numbered from 1 to N), featuring a total of M metro stations (numbered from 0 to $M - 1$) and indicating, for each line, those stations at which the line stops. Please compute the expected number of questions that player B needs to ask to find the answer, in the optimal strategy.

In other words, given a strategy S , note $Q_{S,j}$ the number of questions asked by the strategy if the metro line in the solution is line j . Then, note

$$E_S = \mathbb{E}[Q_S] = \frac{1}{N} \sum_{j=1}^N Q_{S,j}$$

the expected value of $Q_{S,j}$ assuming that j is uniformly chosen from the set of all metro lines. Your task is to compute $\min_S E_S$.

If it is not always possible for player B to know which line player A had in mind with certainty, output `not possible`.

Input

The first line contains the number N . The second line contains the number M . Then follow M lines: the k^{th} such line contains first a positive integer $n \leq N$, then a space, and then n space-separated integers s_1, s_2, \dots, s_n ; these are the metro stations at which line k stops. A line stops at a given station at most once.

Output

The output should contain a single line, consisting of a single number: the minimum expected number of questions that player B must ask in order to find the correct metro line, or `not possible` (in lowercase characters). Answers within 10^{-4} of the correct answer will be accepted.

Limits

- $1 \leq N \leq 18$;
- $1 \leq M \leq 50$.

Sample Input 1

```
5
4
3 0 3 4
3 0 2 3
3 2 3 4
2 1 2
```

Sample Output 1

```
2
```

Sample Input 2

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

Sample Output 2

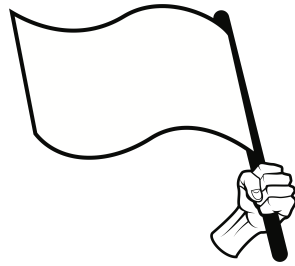
```
1.666666666666667
```

Sample Explanation 2

Ask the first question about station 0: this is optimal by symmetry of the problem. This lets us distinguish between line 1, which stops at station 0, and lines 2 and 3, which do not. If needed, ask a second question to distinguish between lines 2 and 3.

Player B asks one question if the answer is line 1, and two questions otherwise. Thus, the expected number of questions she will ask is $(1 + 2 \times 2)/3$.

D: Flag performance



You are in charge of a team of *sorted gymnastics*. This new discipline involves teams of N members. Each team member dresses with a different colour (a number from 1 to N) and holds a coloured flag. Flags have unique colours, also numbered from 1 to N . A performance consists of exactly K steps. At each step, two members exchange their flags. You are free to choose the initial configuration of the flags. The only constraint is that, at the end of the performance, each participant must hold the flag corresponding to the colour of his outfit.

Being the team captain, you would like the performance to be as unpredictable as possible. You consider T possible initial configurations of flags among the team members, and wonder: in how many ways can the team perform the task for each of these initial configurations?

For each of the given T initial configurations, compute the number of possible ways to do the performance. As the answers may be very large, return them modulo the prime number 1 000 000 007.

Input

Each line consists of space-separated integers. The first input line contains the numbers N , K , and T . Then follow T lines. The k^{th} such line consists of N distinct space-separated integers $c_{k,1}, c_{k,2}, \dots, c_{k,N}$, representing the k^{th} initial configuration of flags among team members. Here, $c_{k,i}$ is colour number of the flag initially in the hands of the team member whose outfit colour is i .

Output

The output should contain T lines. The k^{th} such line should consist of a single number: the number of possible sequences of exchanges that start from the k^{th} configuration and satisfy the constraints listed above, modulo 1 000 000 007.

Limits

- $2 \leq N \leq 30$;
- $1 \leq K \leq 50$;
- $1 \leq T \leq 10\,000$.

Sample Input 1

```
4 2 1
4 1 2 3
```

Sample Output 1

```
0
```

Sample Explanation 1

It is impossible to sort the flags with two exchanges.

Sample Input 2

```
4 3 1
4 1 2 3
```

Sample Output 2

```
16
```

Sample Input 3

```
6 15 10
5 6 1 2 4 3
2 4 1 6 5 3
4 1 3 6 5 2
1 3 2 4 5 6
4 5 6 1 2 3
1 2 5 3 6 4
6 4 2 3 1 5
3 6 4 1 2 5
4 5 1 2 6 3
6 1 4 3 2 5
```

Sample Output 3

```
310571736
0
745108126
996135367
597596468
745108126
0
0
310571736
0
```

E: Nicest view



Paris is so crowded with tourists during the Olympic games! You want to escape the city and go on a hike on a linear trail path. Every kilometre on that trail, including at start and end, is a milestone, on which is written the stone's altitude. The slope between two consecutive stones is constant, and no two stones have the same altitude.

Planning to come back with your friends, you try to identify the point of the hike at which you had the nicest view. The beauty of a point of view is defined as the distance (measured in kilometres) between your position and the previous position, on your hike, that was at the same altitude. If such a previous position fails to exist, it means that you can see the city and its smog, and the beauty of that view is zero.

You have listed the altitudes of the milestones. What is the maximal beauty on your hike?

Input

The input consists of two lines. The first line contains a single integer N , which is the number of milestones on the trail path. The second line contains N space-separated integers H_1, H_2, \dots, H_N ; each integer H_k is the altitude (measured in metres) of the k^{th} milestone on the path.

Output

The output should contain a single line, consisting of a single number S : the best beauty score on your hike. This number is written either as an integer or as an irreducible fraction N/D for which $D \geq 2$; we recall that a fraction N/D is *irreducible* when the greatest common divisor of N and D is 1.

Limits

- $1 \leq N \leq 100\,000$;
- $0 \leq H_k \leq 1\,000\,000$ for all $k \leq N$;
- the integers H_k are pairwise distinct.

Sample Input 1

```
7
0 5 3 1 4 8 2
```

Sample Output 1

```
13/4
```

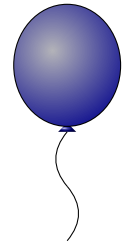
Sample Input 2

```
5  
3 5 8 7 1
```

Sample Output 2

```
0
```

F: Programming-trampoline-athlon!



Programming competitions are fun and exciting. Programming should be an Olympic sport! At least, this is what we believe. However, when we suggested this to some of our friends, they did not seem to share our excitement. So, we decided to suggest a combined sport that will be more interesting to watch. Programming-trampoline-athlon! (we are still working on the name.)

The idea is as follows. This is a team sport, where each team comprises of 3 members. The team has at its disposal 1 hour, 1 computer, and 1 trampoline. At all times, there must be at most one team member using the computer and at least one team member jumping the trampoline. At the beginning of the competition, the team is given 6 programming problems, and 6 trampoline elements (exercises). The team decides how to partition the trampoline elements between its members, such that each team member has to perform 2 of the given elements on the trampoline. The programming tasks are solved cooperatively by the team members, but no one member can spend more than 25 minutes on the computer in total. The scoring is comprised of two parts, which are added together:

- The programming score ranges between 0 and 60. The team receives 10 points for every correctly solved problem.
- The trampoline score ranges between 0 and 40. Each of the 6 trampoline elements receives a score from the judges that ranges between 0 and 10. The final trampoline execution score is determined by dropping the highest and lowest of the 6 scores, and then adding up the remaining 4 scores.

Before we pitch this new sport to the International Olympic Committee, we want everything to be ready in order to show them just how serious we are. Thus, each team should receive a medal when no more than two other teams obtained a strictly higher score. However, in order to cope with a recent shortage of medals, the jury was instructed to make sure that there would be no more than 1 000 teams deserving a medal. We ask you to write a program that determines the medallists, given the performance of the different teams.

Input

The first line contains the number N of competing teams. Then follow N lines. Each of these lines describes a team and contains space-separated values $C, P, E_1, E_2, E_3, E_4, E_5, E_6$; C is a five-letter code used to identify the team, P is an integer specifying the number of problems the team solved, and E_i is an integer specifying the execution score of trampoline element number i .

Output

The output should contain M lines, where M is the number of medallists. Each line should represent a medallist team, by containing two space-separated values C and S , where C is the team code and S is the total score of the team. Medallist teams should be listed by decreasing total score and, in case of ties, by input order.

Limits

- $3 \leq N \leq 100\,000$;
- $3 \leq M \leq 1\,000$;
- C consists of five uppercase English letters;
- $0 \leq P \leq 6$;
- $0 \leq E_i \leq 10$ for all $1 \leq i \leq 6$;
- different teams always have different team codes.

Sample Input 1

```
5
EMAIL 3 5 6 7 8 9 10
CRASH 2 7 1 8 2 8 1
MOUSE 4 0 9 3 9 1 7
SWERC 6 3 1 4 1 5 9
PAINT 6 0 0 0 0 0 10
```

Sample Output 1

```
SWERC 73
EMAIL 60
MOUSE 60
PAINT 60
```

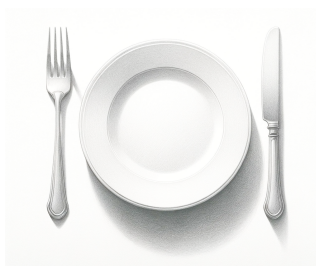
Sample Input 2

```
4
CRAZY 4 0 2 4 6 8 10
JAZZY 2 9 9 9 9 9 9
JUICY 3 2 9 10 9 10 1
FUZZY 5 0 1 1 2 3 5
```

Sample Output 2

```
CRAZY 60
JUICY 60
FUZZY 57
```

G: Favourite dish



France is a country of gastronomy. For a dish, both the taste and plating are important. Nevertheless, when different people evaluate a dish, some focus more on taste and some focus more on plating. At the Olympic Village dining hall, there are N dishes, numbered from 1 to N ; each dish has a score on its taste and a score on its plating. There are also M persons, numbered from 1 to M ; each person has a weight on taste and a weight on plating. One person's final score of a dish is the weighted average of the dish's scores on taste and plating.

The chefs at the Olympics want to provide everyone with their favourite dish on the evening of the closing ceremony. Your task is to calculate everyone's favourite dish. If multiple dishes tie for the highest score as a person's favourite, choose the one with the smallest number.

Input

Each line contains two space-separated integers. The first line contains the numbers N and M . Then follow N lines; the k^{th} such line contains two integers t_k and p_k , which are the scores of the dish k on taste and on plating. Then come M more lines; the ℓ^{th} such line contains two integers T_ℓ and P_ℓ , which are the weights of person ℓ on taste and on plating.

Output

The output should contain M lines. The ℓ^{th} such line should contain one number: the number of the favourite dish of person ℓ .

Limits

- $1 \leq N \leq 500\,000$;
- $1 \leq M \leq 500\,000$;
- $0 \leq t_k \leq 1\,000\,000$, $0 \leq p_k \leq 1\,000\,000$, and $(t_k, p_k) \neq (0, 0)$ for all $k \leq N$;
- $0 \leq T_\ell \leq 1\,000\,000$, $0 \leq P_\ell \leq 1\,000\,000$, and $(T_\ell, P_\ell) \neq (0, 0)$ for all $\ell \leq M$;
- the N pairs (t_k, p_k) are pairwise distinct;
- the M pairs (T_ℓ, P_ℓ) are pairwise distinct.

Sample Input 1

```
4 3
2 5
3 4
4 2
1 6
6 4
2 8
5 5
```


Sample Output 1

```
2
4
1
```

Sample Explanation 1

Here is the score table for each person on each dish. Each person's favourite dish is indicated with a *; person 3 has three tied favourite dishes, so we chose the first one.

		Dish			
		1	2	3	4
Person	1	3.2	3.4*	3.2	3
	2	4.4	3.8	2.4	5*
	3	3.5*	3.5	3	3.5

Sample Input 2

```
3 4
1 0
0 2
0 1
1 1
2 2
2 1
1 0
```

Sample Output 2

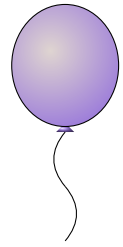
```
2
2
1
1
```

Sample Explanation 2

Here is the score table for each person on each dish. Each person's favourite dish is indicated with a *.

		Dish		
		1	2	3
Person	1	0.5	1*	0.5
	2	0.5	1*	0.5
	3	2/3*	2/3	1/3
	4	1*	0	0

H: Break a leg!



For the first time, breakdance will be featured in the Olympics. And you get to participate! Well, you get to participate to the jury... More precisely, you get to build the table in front of which the jury will be seated: still, that is an impressive feat, congratulations!

Actually, the top of the table is already built: it is plane, has constant width and constant density, and its shape consists in the interior of an N -sided non-crossing polygon $P_1P_2 \dots P_N$ in which no three vertices are collinear (i.e., no line goes through three vertices or more). You have three table legs of same length and negligible width. Your task is to place them at distinct corners of the table so that the table remains stable when standing on these legs. In other words, you must choose three vertices P_i , P_j and P_k of the polygon such that the centre of gravity of the polygon lies in the interior of the triangle $P_iP_jP_k$ (and not on its boundary).

In how many different ways can you do this? If two ways of placing legs differ only by a permutation of the legs, they are not counted as different ways.

Input

The first line contains the number N . Then follow N lines: the i^{th} of these lines contains two space-separated integers x_i and y_i , which are the x -coordinate and the y -coordinate of the vertex P_i .

Output

The output should contain a single line, consisting of a single integer: the number of ways of placing legs such that the table remains stable.

Limits

- $3 \leq N \leq 100\,000$;
- $-1\,000\,000 \leq x_i \leq 1\,000\,000$ and $-1\,000\,000 \leq y_i \leq 1\,000\,000$ for all $i \leq N$;
- whenever $1 \leq i < j < k \leq N$, the vertices P_i , P_j and P_k are not collinear;
- the polygonal shape $P_1P_2 \dots P_N$ is non-crossing.

Sample Input 1

```
4
0 0
1 0
1 1
0 1
```

Sample Output 1

0

Sample Input 2

```
4
0 0
5 0
6 6
0 5
```

Sample Output 2

1

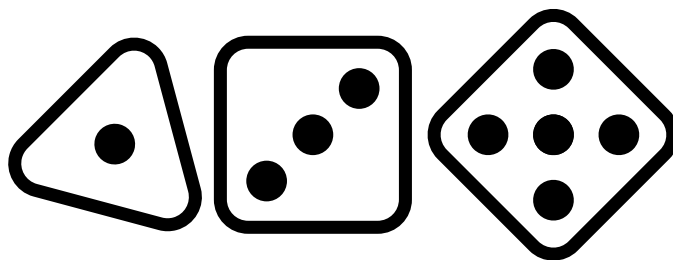
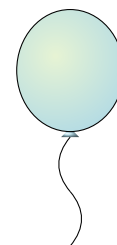
Sample Input 3

```
5
0 0
2 0
2 20
1 1
0 20
```

Sample Output 3

5

I: Throwing dice



Alice and Bob are discussing penalty shoot-outs and their randomness: “We might as well be throwing dice to determine the winner!”, Alice said. And so they started simulating penalty shoot-outs by each throwing dice, summing the points indicated on their dice, and comparing these sums. The player with the largest sum wins; in case both sums are equal, there is a tie.

But even in such situations, some player might have an edge over their opponent, depending on which dice they throw. Thus, just by looking at the dice they are about to throw, Alice and Bob want to determine who has the better edge.

Alice has M fair dice, with A_1, A_2, \dots, A_M sides. For all integers k and ℓ such that $1 \leq k \leq M$ and $1 \leq \ell \leq A_k$, the k^{th} die of Alice has a probability $1/A_k$ of showing its face numbered ℓ . Then, Alice’s score is the sum of the numbers displayed by her M dice. Similarly, Bob has N fair dice, with B_1, B_2, \dots, B_N sides.

Given these dice, Alice has a probability \mathbb{P}_A of having a strictly larger score than Bob, and Bob has a probability \mathbb{P}_B of having a strictly larger score than Alice. Which probability is the largest one?

Input

The input consists of three lines, each one containing space-separated integers. The first line contains the numbers M and N . The second line contains the numbers A_1, A_2, \dots, A_M . The third line contains the numbers B_1, B_2, \dots, B_N .

Output

The output should contain a single line, consisting of a single uppercase word: ALICE if $\mathbb{P}_A > \mathbb{P}_B$, TIED if $\mathbb{P}_A = \mathbb{P}_B$, and BOB if $\mathbb{P}_A < \mathbb{P}_B$.

Limits

- $1 \leq M \leq 100\,000$;
- $1 \leq N \leq 100\,000$;
- $4 \leq A_k \leq 1\,000\,000\,000$ for all $k \leq M$;
- $4 \leq B_k \leq 1\,000\,000\,000$ for all $k \leq N$.

Sample Input 1

```
8 1
4 4 4 4 4 4 4 4
6
```

Sample Output 1

```
ALICE
```

Sample Explanation 1

Since Alice has 8 dice, her score is always 8 or more; Bob's score is always 6 or less. Hence, Alice has a probability $\mathbb{P}_A = 100\%$ of beating Bob, and he has a probability $\mathbb{P}_B = 0\%$ of beating her. Consequently, $\mathbb{P}_A > \mathbb{P}_B$.

Sample Input 2

```
2 2  
6 4  
4 6
```

Sample Output 2

```
TIED
```

Sample Explanation 2

Alice has a probability $\mathbb{P}_A = 125/288$ of beating Bob; he also has a probability $\mathbb{P}_B = 125/288$ of beating her.

J: Olympic goodies



Freshly arrived on the market, retailer YAOGS (Yet Another Olympic Goodies Seller) sells very expensive Olympics-themed items. To make themselves better known to the public, they half-heartedly decide to give away some of these items via a contest: the first person to answer correctly the question “How many circles are there in the Olympic Games logo?” can thus gain up to P very expensive but equally valued items.

To spice things up (and spend less), YAOGS however opts for an additional challenge, as follows. The P available items are positioned along some, but possibly not all of the alleys of YAOGS’s headquarters; each alley can thus contain 0, 1, or more items. For reasons unknown, these alleys form a connected, undirected, acyclic graph (i.e., a tree) with N nodes, numbered from 0 to $N - 1$.

The winner knows N but has no idea about either the tree structure or the items’ placement. Once goodies are placed, her task is to choose a start node m and an end node n . She can then collect all the items on the (unique) path from m to n in the tree.

YAOGS decides to cleverly place the goodies so that they minimise the maximum number of items that can possibly be collected. Assuming they properly carry out this task, what is the maximum number of items the winner can collect?

Input

Each line contains two space-separated integers. The first line contains the numbers N and P . Then follow $N - 1$ lines; the k^{th} such line contains two integers a_k and b_k , meaning that there is an edge between the nodes a_k and b_k of the tree.

Output

The output should contain a single line, consisting of a single integer: the maximum number of items that can be collected by the winner.

Limits

- $1 \leq N \leq 100000$;
- $1 \leq P \leq 100000$;
- $0 \leq a_k \leq N - 1$ and $0 \leq b_k \leq N - 1$ for all $k \leq N - 1$;
- the set of edges in the input file describes a valid tree structure.

Sample Input

```
5 5
0 1
0 2
2 3
2 4
```

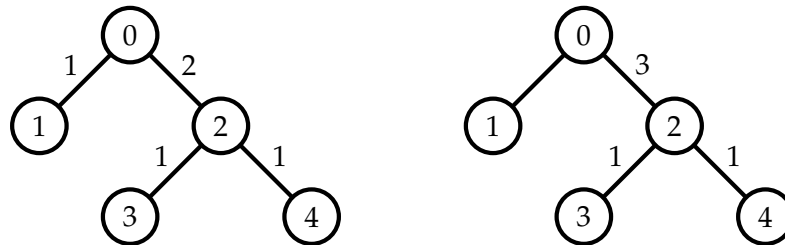
Sample Output

```
4
```

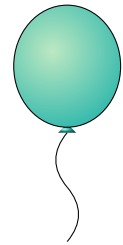
Sample Explanation

For the tree in the sample input, depicted below, an optimal item placement by YAOGS guarantees that the winner cannot collect more than four items.

The figures below show two possible item placements to achieve this optimality. In the first one, the four items may be collected by choosing, for instance, nodes 1 and 3. In the second one, the four items may be collected by choosing, for instance, nodes 0 and 4.



K: Team selection



Two team leaders get to assemble their teams by choosing team members among a set of players that are numbered from 1 to N . The leaders take turns, each picking the k^{th} player among the remaining ones, according to their ideas of which one of the remaining players would be the best addition to their teams.

Given the choices of the two leaders (the first team leader starts first), please compute the list of players in each team.

Input

The input consists of three lines. The first line contains the single integer N . The second line contains $N/2$ space-separated integers $a_1, a_2, \dots, a_{N/2}$ representing the choices of the first team leader: during the $(2k - 1)^{\text{th}}$ turn, the first leader chose the a_k^{th} remaining player. The third line contains $N/2$ space-separated integers $b_1, b_2, \dots, b_{N/2}$ representing the choices of the second team leader: during the $2k^{\text{th}}$ turn, the second leader chose the b_k^{th} remaining player.

Output

The output should contain two lines, each containing $N/2$ space-separated integers. The first line should contain the list $x_1, x_2, \dots, x_{N/2}$ of the players chosen to become members of the first team, in the order they were chosen: the player x_k was chosen during the $(2k - 1)^{\text{th}}$ turn. The second line should contain the list $y_1, y_2, \dots, y_{N/2}$ of the players chosen to become members of the second team, in the order they were chosen: the player y_k was chosen during the $2k^{\text{th}}$ turn.

Limits

- $2 \leq N \leq 4\,000\,000$;
- N is a multiple of 2;
- the choices of the team leaders are valid: at each step, they are between 1 and the number of remaining players (inclusive).

Sample Input

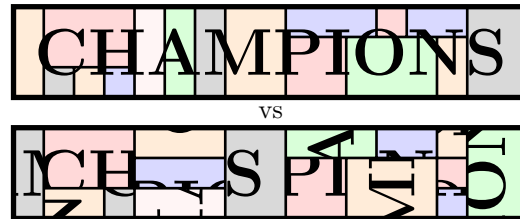
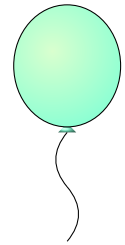
```
4
1 1
2 1
```

Sample Output

```
1 2
3 4
```


Page intentionally left blank

L: Broken trophy



Coming back home after triumphally winning your long-coveted trophy, you discover that it was shattered to pieces in your trunk. It just remains to repair it.

Your trophy had the shape of a rectangle of size $3 \times N$, for some integer $N \geq 1$, thereby consisting of 3 lines and N columns, containing a total of $3N$ unit squares. It was broken into K pieces, the k^{th} piece being a rectangle of size $A_k \times B_k$ for some integers A_k and B_k such that $1 \leq A_k \leq B_k \leq 3$. Such pieces may have been rotated, or even flipped, in the havoc that is your trunk.

As the first step towards repairing your trophy, you should reassemble them in the form of a rectangle of size $3 \times N$. More precisely, you have drawn, on a sheet of paper, a $3 \times N$ rectangle on which you will place your K pieces, and you need to know, for all integers $i \leq 3$ and $j \leq N$, which piece will cover the unit square on the i^{th} line and j^{th} column of your rectangle.

Input

The input consists of three lines, each one containing space-separated integers. The first line contains the numbers K and N . The second line contains the numbers A_1, A_2, \dots, A_K . The third line contains the numbers B_1, B_2, \dots, B_K .

Output

The output should contain three lines, each one consisting of N space-separated integers. If you plan to cover the unit square on the i^{th} line and j^{th} column with the k^{th} piece, the j^{th} number on the i^{th} output line should be the integer k .

In case there are several ways to reassemble your pieces in the form of a rectangle of size $3 \times N$, every output representing one of these ways is considered correct.

Limits

- $1 \leq K \leq 300\,000$;
- $1 \leq N \leq 100\,000$;
- $1 \leq A_k \leq B_k \leq 3$ for all $k \leq K$;
- the pieces described in the input can be reassembled in the form of a rectangle of size $3 \times N$.

Sample Input 1

```
16 17
1 2 1 1 2 1 2 1 1 1 1 1 2 2 1 1
3 3 1 3 2 3 3 1 1 2 2 3 3 3 1 3
```

Sample Output 1

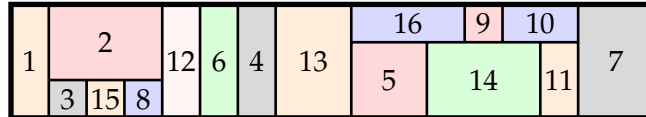
```

1 2 2 2 12 6 4 13 13 16 16 16 9 10 10 7 7
1 2 2 2 12 6 4 13 13 5 5 14 14 14 11 7 7
1 3 15 8 12 6 4 13 13 5 5 14 14 14 11 7 7

```

Sample Explanation 1

This output represents the following reassembling:



Sample Input 2

```

16 17
1 2 1 1 2 1 2 1 1 1 1 1 2 2 1 1
3 3 1 3 2 3 3 1 1 2 2 3 3 3 1 3

```

Sample Output 2

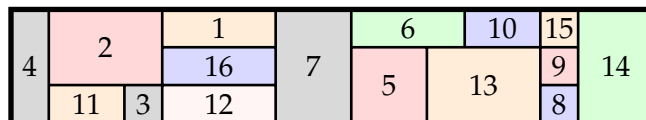
```

4 2 2 2 1 1 1 7 7 6 6 6 10 10 15 14 14
4 2 2 2 16 16 16 7 7 5 5 13 13 13 9 14 14
4 11 11 3 12 12 12 7 7 5 5 13 13 13 8 14 14

```

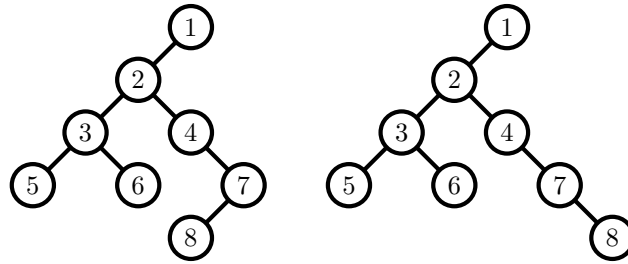
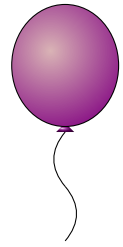
Sample Explanation 2

This output represents the following reassembling:



Both reassemblings are valid, even though sample inputs 1 and 2 are the same.

M: In-order



The opening ceremony for the Olympic Games will take place on the river with teams on boats. The layout of the athletes on top of the boat has been designed in a very specific way: for each team, the N athletes (conveniently numbered from 1 to N) are arranged as a binary tree.

The organiser has also designed the pre-order traversal, post-order traversal, and a (possibly empty) consecutive part of the in-order traversal of the binary tree that each team must follow.

Now, to make sure there are enough tree layouts so that each team can have a distinct one, you are asked to calculate the quantity of different possible in-order traversals, say T , modulo the prime number 999 999 937.

Input

The input consists of four lines. The first line contains the number N . Each subsequent line contains a list of N space-separated integers. The second line contains a list A_1, A_2, \dots, A_N , where A_k is the number of the k^{th} athlete found in pre-order traversal. The third line contains a list B_1, B_2, \dots, B_N , where B_k is the number of the k^{th} athlete found in post-order traversal. The fourth line contains a list C_1, C_2, \dots, C_N , where C_k is either the number of the k^{th} athlete found in in-order traversal, or 0 if the organiser did not say who that k^{th} athlete should be.

Output

The output should contain a single line, consisting of a single integer S : this is the only integer such that $0 \leq S < 999\,999\,937$ and for which $T - S$ is divisible by 999 999 937.

Limits

- $1 \leq N \leq 500\,000$;
- there exists at least one binary tree with such pre-order, post-order and in-order traversals;
- the integers k for which $C_k \geq 1$ form a (possibly empty) sub-interval of the set $\{1, 2, \dots, N\}$; in other words, whenever $k \leq \ell$ and both C_k and C_ℓ are positive, all the integers $C_k, C_{k+1}, \dots, C_\ell$ are positive.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Explanation 1

The graphs given above the problem statement are the two possible binary trees. Their in-order traversals are:

- 5 3 6 2 4 8 7 1
- 5 3 6 2 4 7 8 1

Sample Input 2

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

Sample Output 2

```
4
```

Sample Explanation 2

The four possible in-order traversals are:

- 3 2 1
- 2 3 1
- 1 3 2
- 1 2 3

Sample Input 3

```
4
1 2 3 4
4 3 2 1
0 4 0 0
```

Sample Output 3

```
3
```

Sample Explanation 3

The three possible in-order traversals are:

- 2 4 3 1
- 1 4 3 2
- 3 4 2 1