Problem A. Alice, Bob and Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 megabytes

Alice and Bob have recently played many complex games, so they want to play something simple to relax.

They remembered a very simple game called "Seven", which can be played with just a deck of cards. The traditional rules of Seven are a bit complex, so Alice and Bob are going to play a simplified version of Seven.

The playing cards used in the game include four suits: Spades, Hearts, Clubs, and Diamonds. Each suit has one card with each number from 1 to 13, totaling 52 cards.

The rules of the simplified game are as follows:

1. First, a deck of 52 cards is evenly split into two parts, with each part containing 26 cards, and Alice and Bob each take one of the parts;
2. The player who has the 7 of Spades places their 7 of Spades on the table;
3. Then, starting with the other player, they take turns playing. In each round, a player must choose one card from their hand to place on the table. The card that can be played must satisfy one of the following two conditions:
   - The card's point value is 7;
   - There is a card on the table that has the same suit as the card to be played, and the point value differs by exactly 1.
4. If both players have played all their cards, the game is a draw. Otherwise, if the player whose turn it is cannot play a card according to the rules, this player loses, and the other player wins.

Since the sequences of different suits in this game are independent, players often align different suits of 7 vertically and connect the sequences on either side of the 7, arranging the same suit cards in a row.

Given that Alice and Bob are both smart enough, now that Alice and Bob's initial hands are provided, please calculate the result of the game when both play optimally.

Input

Since one player's hand can be used to infer the other player's hand, the input only contains all of Alice's cards.

The input consists of a single line with 26 space-separated strings, each representing a card in Alice's hand. The cards are all distinct, and can be given in any order.

Each card is described by its suit and point value. The suit is represented by a character: `S` for Spades, `H` for Hearts, `C` for Clubs, and `D` for Diamonds. The point value is represented by a corresponding one or two-digit number, with the minimum being 1 and the maximum being 13, without leading zeros. For example, the 7 of Spades corresponds to the string `S7`.

Output

The output consists of a single line with a string that indicates the result of the game. If Alice will win this game when both players play optimally, output `Alice`; if Bob will win, output `Bob`; if it's a draw, output `Draw`.

Examples

standard input
S4 S6 S11 S13 H3 H4 H5 H6 H8 H10 H12 C2 C4 C5 C6 C8 C9 C11 C13 D1 D4 D5 D6 D8 D10 D12
standard output
Bob

standard input
S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13
standard output
Draw

standard input
H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 C13 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13
standard output
Alice

Problem B. Quick Trip

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Bus route 322 travels along a linear route consisting of n stops, numbered from 1 to n. Tickets for boarding the bus are sold only in advance, before the bus departs from the starting stop. Each ticket is marked with the stop number where the passenger will board and the stop number where the passenger will leave the bus. This information, which does not change during the trip, is displayed on the driver's monitor. The driver stops at a stop only if the monitor shows passengers who are boarding or leaving (or both) at that stop.

On the evening trip, the driver is tired and wants to get home quickly. Unfortunately for the passengers, he acts as follows: he only looks at the monitor (ignoring whether there are waiting passengers at the stops or passengers on the bus who need to get off) and stops at a stop only when the monitor shows both passengers who are boarding and passengers who are leaving the bus at that stop. As a result, some waiting passengers may not be able to board, and some passengers on the bus may not be able to get off at their desired stop.

At the starting stop, the bus picks up all waiting passengers. The bus must stop at the last stop where all remaining passengers leave. There are no waiting passengers at the last stop. At each stop, the bus has enough space to pick up all waiting passengers.

Write a program that determines the total number of passengers who bought tickets but were unable to board the bus, as well as the total number of passengers who boarded the bus but were unable to get off at their desired stop.

Input

The first line of standard input contains two integers n and m ($2 \le n \le 100,000$, $2 \le m \le 100,000$), separated by a space. These numbers represent the number of stops and the number of passengers who bought tickets for the last bus trip. The next m lines contain two integers a and b ($1 \le a < b \le n$), separated by a space — the stop number where the passenger will wait to board and the stop number where the passenger wants to leave. This data is sorted by the boarding stop numbers.

Output

Your program should output two lines, each containing one integer. The first line should contain the number of passengers who were unable to board the bus. The second line should contain the number of passengers who boarded the bus but were unable to get off at their desired stop.

Example

standard input	standard output
8 5 1 2 1 3 1 3 2 5 4 7	1 3

Problem C. Corrupted Order

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	1024 megabytes

This world is in imminent danger! Order has completely fallen into chaos.

Order can be abstracted as an $n \times n$ matrix, where the matrix contains a permutation of numbers from 1 to n^2. You want to save the world, so you called upon a deity to help restore order. However, the deity is not omnipotent; it can only swap two numbers in the same row or the same column of the matrix. Moreover, it does not know how to swap to restore order, and must rely on your guidance.

Fortunately, you do not necessarily need to complete the restoration in the minimum number of swaps. You only need to ensure that your number of swaps is not worse than the worst-case scenario. In other words, if your number of swaps is k, and the maximum minimum number of swaps for all permutations from 1 to n^2 is k_0, you only need to satisfy $k \le k_0$.

Restoration refers to transforming the matrix into the following matrix:

$$\begin{matrix} 1 & 2 & 3 & \cdots & n \\ n+1 & n+2 & n+3 & \cdots & 2n \\ 2n+1 & 2n+2 & 2n+3 & \cdots & 3n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (n-1)n+1 & (n-1)n+2 & (n-1)n+3 & \cdots & n^2 \end{matrix}$$

Input

The first line of input contains a positive integer n ($1 \le n \le 1000$).

Each of the next n lines contains n positive integers. Together, they represent the $n \times n$ matrix. It is guaranteed that each number from 1 to n^2 appears in the matrix exactly once.

Output

The first line should contain a non-negative integer k: the number of swaps you made. This number should not be greater than the minimum number of swaps for the worst possible case of $n \times n$ matrix.

The next k lines should each contain four positive integers: x_1, y_1, x_2, y_2. They indicate that you swapped the number in row x_1, column y_1 with the number in row x_2, column y_2.

You need to ensure that $x_1 = x_2$ or $y_1 = y_2$.

If there is more than one solution, print any one of them.

Examples

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

Note

For Sample 1, it can be proven that this is one of the solutions with the minimum number of swaps, and it clearly meets the conditions.

For Sample 2, the sample output's solution is not the one with the minimum number of swaps, but we know that there exists a permutation from 1 to n^2 (the previous example) that requires at least 3 swaps, so this solution is also feasible.

For Sample 3, we allow the case where $(x_1, y_1) = (x_2, y_2)$.

For Sample 4, note that k can be equal to 0.

Problem D. Valeria and Pencils

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Valeria loves colored pencils. To protect the sharp tips from breaking, she puts a cap on the tip of each pencil. She has a set of n different colored pencils and m caps for each color. All the caps are stored together in one bag.

Initially, none of Valeria's pencils are capped. She will repeatedly perform the following operation:

- Randomly draw one cap from the bag with equal probability. Let this cap be of color i.
- If the pencil of color i does not have a cap yet, cap it with this cap.
- Otherwise, discard the cap (without returning it to the bag).

Calculate the probability that after t operations all the pencils will be capped, modulo 998 244 353.

Input

The input consists of three positive integers: the number of colors of pencils n ($1 \le n < 998\,244\,353$), the number of caps for each color m ($1 \le m < 998\,244\,353$), and the number of operations t ($1 \le t \le m \cdot n$). It is additionally guaranteed that $1 < m \cdot n < 998\,244\,353$ and $1 \le T \times N \le 10^5$.

Output

It can be shown that:

- The probability will always be a rational number.
- Given the constraints of the problem, the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where the denominator q will not be divisible by $998, 244, 353$.
- There exists exactly one integer x in $[0, 998, 244, 353)$ such that $q \cdot x \equiv p \pmod{998, 244, 353}$, and $0 \le x < 998, 244, 353$.

As the answer, output the corresponding value of x.

Examples

standard input	standard output
`3 2 2`	`0`
`3 2 3`	`199648871`

Note

The probability can be computed as follows: we consider the number A of different sequences of T operations such that all pencils have caps at the end, then we consider the total number B of sequences of T operations, then the probability is A/B. Note that even if two caps have the same color, they are considered different in these calculations.

Problem E. Edges and Divisors

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 megabytes

We play a game on a directed acyclic graph $G = (V, E)$. We start at vertex $s_1 \in V$. The rules of the game are as follows:

- The game consists of rounds numbered from 1.
- In the i-th round, the player selects a path p_i starting from s_i and containing at least one edge such that the sum of the weights of all edges belonging to this path is an exact multiple of $(i + 1)$. If the player cannot select such a path, the player has failed, and will not score any points. Otherwise, the round ends successfully, and the endpoint of p_i is recorded as s_{i+1}.
- After a successful round, the player can either end the game or continue with the next round. If the player chooses to end the game, the selected i paths $p_1, \ldots, p_i$ are called doubling paths, and the score is calculated.

If the player has not failed, then when ending the game, for the selected doubling paths $p_1, \ldots, p_k$, the score of the game is defined as $\sum_{i=1}^{k} a_i \, |p_i| \, /k$, where $|p_i|$ represents the number of edges in path p_i, and a_i is the weight given in the input. Clearly, as the graph is acyclic, at most $(n - 1)$ paths can be selected, so the input only provides the weights $a_1, \ldots, a_{n-1}$.

Given the graph and the starting vertex, calculate the maximum achievable score in the game.

Input

The first line of input contains three integers, n, m, and s_1: the number of vertices, the number of edges, and the index of the starting vertex ($2 \le n \le 100$; $1 \le m \le \frac{n(n-1)}{2}$; $1 \le s_1 \le n$).

The second line contains $(n - 1)$ integers $a_1, \ldots, a_{n-1}$: the weights used to calculate the score ($1 \le a_1 \le a_2 \le \ldots \le a_{n-1} \le 10^9$).

Each of the next m lines contains three integers, u_i, v_i, and w_i, describing a directed edge from u_i to v_i with weight w_i ($1 \le u_i, v_i \le n$; $u_i \ne v_i$; $1 \le w_i \le 10^9$). It is guaranteed that the graph is acyclic, the graph is connected if we treat edges as undirected, and there are no multiple edges.

Output

Output a single line containing two integers separated by a space.

If at least one path can be selected, there exists an optimal selection scheme that maximizes the score, and the optimal score can be represented as p/q where p and q are coprime integers and $q > 0$. In this case, output p and q. Otherwise (if it is impossible to select any paths), output "`-1 -1`".

Examples

standard input	standard output
5 5 1 1 11 21 1211 1 2 4 1 3 11 2 5 9 3 4 12 4 5 13	6 1
9 16 1 1 10 100 1000 10000 100000 1000000 10000000 1 2 2 1 3 3 2 3 5 2 5 7 3 4 11 3 5 13 3 6 17 3 7 19 4 7 23 5 6 29 5 8 31 6 7 37 6 8 41 6 9 43 7 9 47 8 9 53	221 3

Note

The selected doubling paths are $p_1 = ((1,2))$ and $p_2 = ((2,5))$.

Problem F. Gryffindor is Gathering Alumni

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The Gryffindor faculty administration is planning to organize a reunion for alumni. They have managed to obtain the contacts of n alumni. Professor McGonagall knows that not all alumni will attend, and she wants to maximize the number of attendees.

Initially, all n alumni received invitations, and each of them responded whether they would attend the reunion or not. McGonagall can influence the situation as follows: she knows that if an alumnus has information that one of their friends will attend the reunion, they will accept the invitation; if they do not see anyone from their friends on the list, they will decline. McGonagall remembers who among the alumni were friends (if a was friends with b, then b was friends with a).

McGonagall can publish the current lists of those who have accepted the invitation. After that, each alumnus will determine their decision based on the published lists (i.e., they will decline if none of their friends are on the list, or accept the invitation if at least one of their friends is on the list). Some declines may change to acceptances and vice versa.

McGonagall can perform this action an arbitrary number of times, each time publishing the state of the responses after the previous mailing, or she can stop the process—then the decisions made based on the last published information (or those initially accepted if nothing was published) become final.

Your task is to determine the maximum number of alumni that can attend the reunion with the optimal choice of strategy by the faculty administration.

Input

The first line of input contains a single integer n ($1 \le n \le 3 \cdot 10^5$) — the number of alumni.

The second line contains n integers, separated by spaces — the initial responses of the alumni (1: the alumnus accepted the invitation, 0: the alumnus declined the invitation) listed in the order in which the alumni are numbered.

The third line contains a single integer m — the number of pairs of friends ($0 \le m \le 4 \cdot 10^5$).

Each of the following m lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$) — the numbers of the alumni who are friends with each other. It is guaranteed that each pair appears no more than once.

Output

One integer: the maximum possible number of attendees.

Example

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

Problem G. Generate Optimal Key

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 megabytes

A *pattern* is defined as a binary string of length L. Given n patterns, we want to find a *key* which is also a binary string of length L.

For a given pattern and a given key, the *error value* is defined as the number of positions where the key differs from the pattern. For example, if the pattern is `101`, and the key is `000`, then the first and third positions are different, so the error value is 2.

We want to find a key such that the sum of the error values for this key and the given n patterns is minimized. Additionally, there are m distinct forbidden keys, and the key we find cannot be one of the forbidden keys.

Input

The first line of input contains three integers, n, m, and L ($1 \le n \le 1000$; $1 \le m \le \min(1000, 2^L - 1)$; $1 \le L \le 1000$).

Each of the following n lines represents a pattern and contains a binary string of length L.

Each of the following m lines represents a forbidden key and contains a binary string of length L. All forbidden keys are distinct.

Output

Output a single integer: the minimum sum of error values for a non-forbidden key and the n given patterns if we select the key optimally.

Examples

standard input	*standard output*
4 1 4 0000 1000 1100 1010 1000	5
2 4 4 0000 0000 0000 1000 0100 0010	2

Problem H. Additional Classes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In the additional math classes, the teacher asked a student to write down three rows of digits of the same length, one below the other:

```
2642
0982
3564
```

After that, the teacher asked the student: "How many ways are there to delete columns such that the sum of the first row and the second row equals the third row?"

In the example above, one can:

- Delete the first three columns ($2 + 2 = 4$)
- Delete only the third column ($262 + 92 = 354$)
- Delete the third and last column ($26 + 9 = 35$)
- Delete all columns ($0 + 0 = 0$)

Thus, the answer will be 4.

Then the teacher randomly selected a digit on the board several times, replaced the selected digit with another, and after each such change, asked the student a similar question.

Help the student answer the teacher's questions.

Input

The first line of the input contains two integers n and q: the number of digits in each row and the number of modifications/questions ($1 \le n \le 10^5$, $0 \le q \le 10^5$).

Each of the next three lines represents a string consisting of exactly n decimal digits. Each of the following q lines contains a description of one query: three integers i, j, and c, indicating that the digit in row i and column j is changed to digit c ($1 \le i \le 3$, $1 \le j \le n$, $0 \le c \le 9$).

Output

Output $q + 1$ integers, one per line. The first line should contain the remainder of the answer before the first modification when divided by $998\,244\,353$. The other q lines should contain the remainder of the answer to the teacher's question after the corresponding modification when divided by $998\,244\,353$.

Example

standard input	standard output
4 1 2642 0982 3564 1 1 3	4 4

Problem I. Arithmetic and Arrays

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given an array of n integers. You are also given an integer x. You can apply the following operation no more than x times — choose any element of the array and increase it by 1.

You need to maximize the sum of the k-th powers of the elements of the array using these operations, and also determine the minimum number of operations required to achieve this maximum.

In this problem, it is considered that $0^0 = 0$.

Input

The first line of input contains three integers n, x, and k ($1 \le n \le 10^5$, $1 \le x \le 10^9$, $0 \le k \le 2$). The second line contains n integers a_i ($-10^5 \le a_i \le 10^5$).

Output

Output two integers — the maximum value of the sum of the k-th powers of the elements of the array and the minimum number of operations, not exceeding x, required to achieve the corresponding sum value.

Examples

standard input	standard output
3 1 2 -5 3 -4	57 1
3 0 0 0 0 0	0 0

Note

In Sample 1, we have two negative integers and one positive, and we need to maximize the sum of the squares. Because for the negative integer a $(a+1)^2 < a^2$, and for positive integer a $(a+1)^2 > a^2$, the maximum is reached by adding 1 to the middle element, and the sum become $(-5)^2 + 4^2 + (-4)^2 = 25 + 16 + 16 = 57$.