Educational Codeforces Round 107 (Rated for Div. 2)

The problem statement has recently been changed. View the changes.

A. Review Site

2 seconds, 256 megabytes

You are an upcoming movie director, and you have just released your first movie. You have also launched a simple review site with two buttons to press — upvote and downvote.

However, the site is not so simple on the inside. There are two servers, each with its separate counts for the upvotes and the downvotes.

n reviewers enter the site one by one. Each reviewer is one of the following types:

- type 1: a reviewer has watched the movie, and they like it — they press the upvote button;
- type 2: a reviewer has watched the movie, and they dislike it — they press the downvote button;
- type 3: a reviewer hasn't watched the movie — they look at the current number of upvotes and downvotes of the movie on the server they are in and decide what button to press. If there are more downvotes than upvotes, then a reviewer downvotes the movie. Otherwise, they upvote the movie.

Each reviewer votes on the movie exactly once.

Since you have two servers, you can actually manipulate the votes so that your movie gets as many upvotes as possible. When a reviewer enters a site, you know their type, and you can send them either to the first server or to the second one.

What is the maximum total number of upvotes you can gather over both servers if you decide which server to send each reviewer to?

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases.

Then the descriptions of t testcases follow.

The first line of each testcase contains a single integer n ($1 \le n \le 50$) — the number of reviewers.

The second line of each testcase contains n integers $r_1, r_2, \dots, r_n$ ($1 \le r_i \le 3$) — the types of the reviewers in the same order they enter the site.

Output

For each testcase print a single integer — the maximum total number of upvotes you can gather over both servers if you decide which server to send each reviewer to.

input
```
4
1
2
3
1 2 3
5
1 1 1 1 1
3
3 3 2
```

output
```
0
2
5
2
```

In the first testcase of the example you can send the only reviewer to either of the servers — they'll downvote anyway. The movie won't receive any upvotes.

In the second testcase of the example you can send all reviewers to the first server:

- the first reviewer upvotes;
- the second reviewer downvotes;
- the last reviewer sees that the number of downvotes is not greater than the number of upvotes — upvote themselves.

There are two upvotes in total. Alternatevely, you can send the first and the second reviewers to the first server and the last reviewer — to the second server:

- the first reviewer upvotes on the first server;
- the second reviewer downvotes on the first server;
- the last reviewer sees no upvotes or downvotes on the second server — upvote themselves.

B. GCD Length

2 seconds, 256 megabytes

You are given three integers a, b and c.

Find two positive integers x and y ($x > 0$, $y > 0$) such that:

- the decimal representation of x without leading zeroes consists of a digits;
- the decimal representation of y without leading zeroes consists of b digits;
- the decimal representation of $gcd(x, y)$ without leading zeroes consists of c digits.

$gcd(x, y)$ denotes the greatest common divisor (GCD) of integers x and y.

Output x and y. If there are multiple answers, output any of them.

Input

The first line contains a single integer t ($1 \le t \le 285$) — the number of testcases.

Each of the next t lines contains three integers a, b and c ($1 \le a, b \le 9$, $1 \le c \le min(a, b)$) — the required lengths of the numbers.

It can be shown that the answer exists for all testcases under the given constraints.

Additional constraint on the input: all testcases are different.

Output

For each testcase print two positive integers — x and y ($x > 0$, $y > 0$) such that

- the decimal representation of x without leading zeroes consists of a digits;
- the decimal representation of y without leading zeroes consists of b digits;

- the decimal representation of $gcd(x, y)$ without leading zeroes consists of c digits.

input

```
4
2 3 1
2 2 2
6 6 2
1 1 1
```

output

```
11 492
13 26
140133 160776
1 1
```

In the example:

1. $gcd(11, 492) = 1$
2. $gcd(13, 26) = 13$
3. $gcd(140133, 160776) = 21$
4. $gcd(1, 1) = 1$

C. Yet Another Card Deck

2 seconds, 256 megabytes

You have a card deck of n cards, numbered from top to bottom, i. e. the top card has index 1 and bottom card — index n. Each card has its color: the i-th card has color a_i.

You should process q queries. The j-th query is described by integer t_j. For each query you should:

- find the highest card in the deck with color t_j, i. e. the card with minimum index;
- print the position of the card you found;
- take the card and place it on top of the deck.

Input

The first line contains two integers n and q ($2 \le n \le 3 \cdot 10^5$; $1 \le q \le 3 \cdot 10^5$) — the number of cards in the deck and the number of queries.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 50$) — the colors of cards.

The third line contains q integers $t_1, t_2, \dots, t_q$ ($1 \le t_j \le 50$) — the query colors. It's guaranteed that **queries ask only colors that are present in the deck**.

Output

Print q integers — the answers for each query.

input

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

output

```
5 2 3 1 5
```

Description of the sample:

1. the deck is $[2, 1, 1, 4, \underline{3}, 3, 1]$ and the first card with color $t_1 = 3$ has position 5;
2. the deck is $[3, \underline{2}, 1, 1, 4, 3, 1]$ and the first card with color $t_2 = 2$ has position 2;
3. the deck is $[2, 3, \underline{1}, 1, 4, 3, 1]$ and the first card with color $t_3 = 1$ has position 3;
4. the deck is $[\underline{1}, 2, 3, 1, 4, 3, 1]$ and the first card with color $t_4 = 1$ has position 1;
5. the deck is $[1, 2, 3, 1, \underline{4}, 3, 1]$ and the first card with color $t_5 = 4$ has position 5.

D. Min Cost String

2 seconds, 256 megabytes

Let's define the *cost* of a string s as the number of index pairs i and j ($1 \le i < j < |s|$) such that $s_i = s_j$ and $s_{i+1} = s_{j+1}$.

You are given two positive integers n and k. Among all strings with length n that contain only the first k characters of the Latin alphabet, find a string with minimum possible *cost*. If there are multiple such strings with minimum *cost* — find any of them.

Input

The only line contains two integers n and k ($1 \le n \le 2 \cdot 10^5; 1 \le k \le 26$).

Output

Print the string s such that it consists of n characters, each its character is one of the k first Latin letters, and it has the minimum possible *cost* among all these strings. If there are multiple such strings — print any of them.

input

```
9 4
```

output

```
aabacadbb
```

input

```
5 1
```

output

```
aaaaa
```

input

```
10 26
```

output

```
codeforces
```

E. Colorings and Dominoes

3 seconds, 512 megabytes

You have a large rectangular board which is divided into $n \times m$ cells (the board has n rows and m columns). Each cell is either white or black.

You paint each white cell either red or blue. Obviously, the number of different ways to paint them is 2^w, where w is the number of white cells.

After painting the white cells of the board, you want to place the maximum number of dominoes on it, according to the following rules:

- each domino covers two adjacent cells;
- each cell is covered by at most one domino;
- if a domino is placed horizontally (it covers two adjacent cells in one of the rows), it should cover only red cells;
- if a domino is placed vertically (it covers two adjacent cells in one of the columns), it should cover only blue cells.

Let the *value* of the board be the maximum number of dominoes you can place. Calculate the sum of *values* of the board over all 2^w possible ways to paint it. Since it can be huge, print it modulo $998\,244\,353$.

Input

The first line contains two integers n and m ($1 \le n, m \le 3 \cdot 10^5$; $nm \le 3 \cdot 10^5$) — the number of rows and columns, respectively.

Then n lines follow, each line contains a string of m characters. The j-th character in the i-th string is * if the j-th cell in the i-th row is black; otherwise, that character is o.

Output

Print one integer — the sum of *values* of the board over all 2^w possible ways to paint it, taken modulo $998\,244\,353$.

input
```
3 4
**oo
oo*o
**oo
```
output
```
144
```

input
```
3 4
**oo
oo**
**oo
```
output
```
48
```

input
```
2 2
oo
o*
```
output
```
4
```

input
```
1 4
oooo
```
output
```
9
```

F. Chainword

3 seconds, 256 megabytes

A chainword is a special type of crossword. As most of the crosswords do, it has cells that you put the letters in and some sort of hints to what these letters should be.

The letter cells in a chainword are put in a single row. We will consider chainwords of length m in this task.

A hint to a chainword is a sequence of segments such that the segments don't intersect with each other and cover all m letter cells. Each segment contains a description of the word in the corresponding cells.

The twist is that there are actually two hints: one sequence is the row above the letter cells and the other sequence is the row below the letter cells. When the sequences are different, they provide a way to resolve the ambiguity in the answers.

You are provided with a dictionary of n words, each word consists of lowercase Latin letters. All words are pairwise distinct.

An instance of a chainword is the following triple:

- a string of m lowercase Latin letters;
- the first hint: a sequence of segments such that the letters that correspond to each segment spell a word from the dictionary;
- the second hint: another sequence of segments such that the letters that correspond to each segment spell a word from the dictionary.

Note that the sequences of segments don't necessarily have to be distinct.

Two instances of chainwords are considered different if they have different strings, different first hints **or** different second hints.

Count the number of different instances of chainwords. Since the number might be pretty large, output it modulo $998\,244\,353$.

Input

The first line contains two integers n and m ($1 \le n \le 8$, $1 \le m \le 10^9$) — the number of words in the dictionary and the number of letter cells.

Each of the next n lines contains a word — a non-empty string of no more than 5 lowercase Latin letters. All words are pairwise distinct.

Output

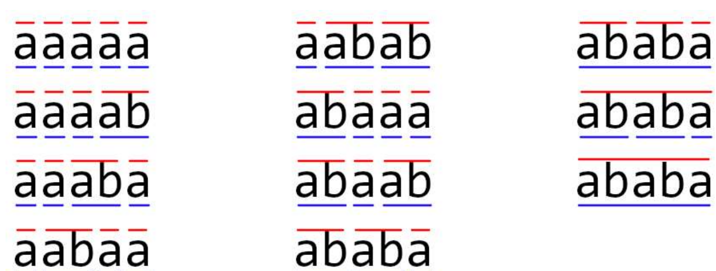
Print a single integer — the number of different instances of chainwords of length m for the given dictionary modulo $998\,244\,353$.

input
```
3 5
ababa
ab
a
```
output
```
11
```

input
```
2 4
ab
cd
```
output
```
4
```

input
```
5 100
a
aa
aaa
aaaa
aaaaa
```
output
```
142528942
```

Here are all the instances of the valid chainwords for the first example:



The red lines above the letters denote the segments of the first hint, the blue lines below the letters denote the segments of the second hint.

In the second example the possible strings are: "abab", "abcd", "cdab" and "cdcd". All the hints are segments that cover the first two letters and the last two letters.

G. Chips on a Board

5 seconds, 512 megabytes

Alice and Bob have a rectangular board consisting of n rows and m columns. **Each row contains exactly one chip.**

Alice and Bob play the following game. They choose two integers l and r such that $1 \le l \le r \le m$ and cut the board in such a way that only the part of it between column l and column r (inclusive) remains. So, all columns to the left of column l and all columns to the right of column r no longer belong to the board.

After cutting the board, they move chips on the remaining part of the board (the part from column l to column r). They make alternating moves, and the player which cannot make a move loses the game. The first move is made by Alice, the second — by Bob, the third — by Alice, and so on. During their move, the player must choose one of the chips from the board and move it any positive number of cells to the left (so, if the chip was in column i, it can move to any column $j < i$, and the chips in the leftmost column cannot be chosen).

Alice and Bob have q pairs of numbers L_i and R_i. For each such pair, they want to determine who will be the winner of the game if $l = L_i$ and $r = R_i$. Note that these games should be considered independently (they don't affect the state of the board for the next games), and both Alice and Bob play optimally.

Input

The first line contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$) — the number of rows and columns on the board, respectively.

The second line contains n integers $c_1, c_2, \dots, c_n$ ($1 \le c_i \le m$), where c_i is the index of the column where the chip in the i-th row is located (so, the chip in the i-th row is in the c_i-th column).

The third line contains one integer q ($1 \le q \le 2 \cdot 10^5$).

Then q lines follow, the i-th of them contains two integers L_i and R_i ($1 \le L_i \le R_i \le m$).

Output

Print a string of q characters. The i-th character should be A if Alice wins the game with $l = L_i$ and $r = R_i$, or B if Bob wins it.

input

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

output

```
BAAAAAB
```

Codeforces (c) Copyright 2010-2021 Mike Mirzayanov
The only programming contests Web 2.0 platform