A. Rainbow Dash, Fluttershy and Chess Coloring

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

One evening Rainbow Dash and Fluttershy have come up with a game. Since the ponies are friends, they have decided not to compete in the game but to pursue a common goal.

The game starts on a square flat grid, which initially has the outline borders built up. Rainbow Dash and Fluttershy have flat square blocks with size 1×1, Rainbow Dash has an infinite amount of light blue blocks, Fluttershy has an infinite amount of yellow blocks.

The blocks are placed according to the following rule: each newly placed block must touch the built on the previous turns figure by a side (note that the outline borders of the grid are built initially). At each turn, one pony can place any number of blocks of her color according to the game rules.

Rainbow and Fluttershy have found out that they can build patterns on the grid of the game that way. They have decided to start with something simple, so they made up their mind to place the blocks to form a **chess coloring**. Rainbow Dash is well-known for her speed, so she is interested in the minimum number of turns she and Fluttershy need to do to get a chess coloring, covering the whole grid with blocks. Please help her find that number!

Since the ponies can play many times on different boards, Rainbow Dash asks you to find the minimum numbers of turns for several grids of the games.

The chess coloring in two colors is the one in which each square is neighbor by side only with squares of different colors.

Input

The first line contains a single integer T ($1 \le T \le 100$): the number of grids of the games.

Each of the next T lines contains a single integer n ($1 \le n \le 10^9$): the size of the side of the grid of the game.

Output

For each grid of the game print the minimum number of turns required to build a chess coloring pattern out of blocks on it.

Example

input

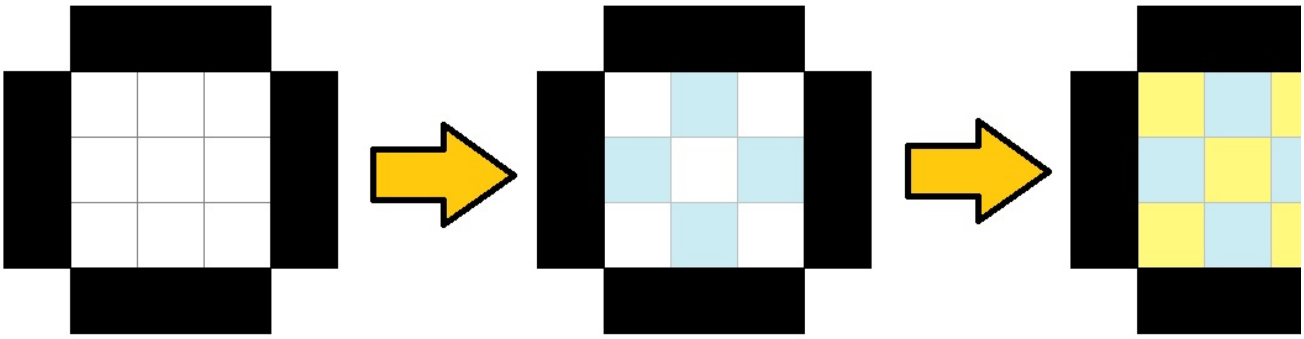
```
2
3
4
```

output

```
2
3
```

Note

For 3×3 grid ponies can make two following moves:



B. Applejack and Storages

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This year in Equestria was a year of plenty, so Applejack has decided to build some new apple storages. According to the advice of the farm designers, she chose to build two storages with non-zero area: one in the shape of a square and another one in the shape of a rectangle (which possibly can be a square as well).

Applejack will build the storages using planks, she is going to spend exactly one plank on each side of the storage. She can get planks from her friend's company. Initially, the company storehouse has n planks, Applejack knows their lengths. The company keeps working so it receives orders and orders the planks itself. Applejack's friend can provide her with information about each operation. For convenience, he will give her information according to the following format:

- $+\ x$: the storehouse received a plank with length x
- $-\ x$: one plank with length x was removed from the storehouse (it is guaranteed that the storehouse had some planks with length x).

Applejack is still unsure about when she is going to order the planks so she wants to know if she can order the planks to build rectangular and square storages out of them after every event at the storehouse. Applejack is busy collecting apples and she has completely no time to do the calculations so she asked you for help!

We remind you that all four sides of a square are equal, and a rectangle has two pairs of equal sides.

Input

The first line contains a single integer n ($1 \le n \le 10^5$): the initial amount of planks at the company's storehouse, the second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$): the lengths of the planks.

The third line contains a single integer q ($1 \le q \le 10^5$): the number of events in the company. Each of the next q lines contains a description of the events in a given format: the type of the event (a symbol $+$ or $-$) is given first, then goes the integer x ($1 \le x \le 10^5$).

Output

After every event in the company, print "YES" if two storages of the required shape can be built from the planks of that company's set, and print "NO" otherwise. You can print each letter in any case (upper or lower).

Example

input

```
6
1 1 1 2 1 1
6
+ 2
+ 1
- 1
+ 2
- 1
+ 2
```

output

```
NO
YES
NO
NO
NO
YES
```

Note

After the second event Applejack can build a rectangular storage using planks with lengths 1, 2, 1, 2 and a square storage using planks with lengths 1, 1, 1, 1.

After the sixth event Applejack can build a rectangular storage using planks with lengths 2, 2, 2, 2 and a square storage using planks with lengths 1, 1, 1, 1.

C. Pinkie Pie Eats Patty-cakes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pinkie Pie has bought a bag of patty-cakes with different fillings! But it appeared that not all patty-cakes differ from one another with filling. In other words, the bag contains some patty-cakes with the same filling.

Pinkie Pie eats the patty-cakes one-by-one. She likes having fun so she decided not to simply eat the patty-cakes but to try not to eat the patty-cakes with the same filling way too often. To achieve this she wants the minimum distance between the eaten with the same filling to be the largest possible. Herein Pinkie Pie called the distance between two patty-cakes the number of eaten patty-cakes strictly between them.

Pinkie Pie can eat the patty-cakes in any order. She is impatient about eating all the patty-cakes up so she asks you to help her to count the greatest minimum distance between the eaten patty-cakes with the same filling amongst all possible orders of eating!

Pinkie Pie is going to buy more bags of patty-cakes so she asks you to solve this problem for several bags!

Input

The first line contains a single integer T ($1 \le T \le 100$): the number of bags for which you need to solve the problem.

The first line of each bag description contains a single integer n ($2 \le n \le 10^5$): the number of patty-cakes in it. The second line of the bag description contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$): the information of patty-cakes' fillings: same fillings are defined as same integers, different fillings are defined as different integers. It is guaranteed that each bag contains at least two patty-cakes with the same filling.

It is guaranteed that the sum of n over all bags does not exceed 10^5.

Output

For each bag print in separate line one single integer: the largest minimum distance between the eaten patty-cakes with the same filling amongst all possible orders of eating for that bag.

Example

input

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

output

```
3
2
0
4
```

Note

For the first bag Pinkie Pie can eat the patty-cakes in the following order (by fillings): 1, 6, 4, 7, 1, 6, 4 (in this way, the minimum distance is equal to 3).

For the second bag Pinkie Pie can eat the patty-cakes in the following order (by fillings): $1, 4, 6, 7, 4, 1, 6, 4$ (in this way, the minimum distance is equal to 2).

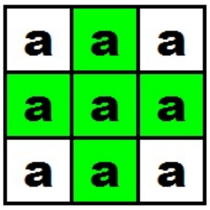
D. Rarity and New Dress

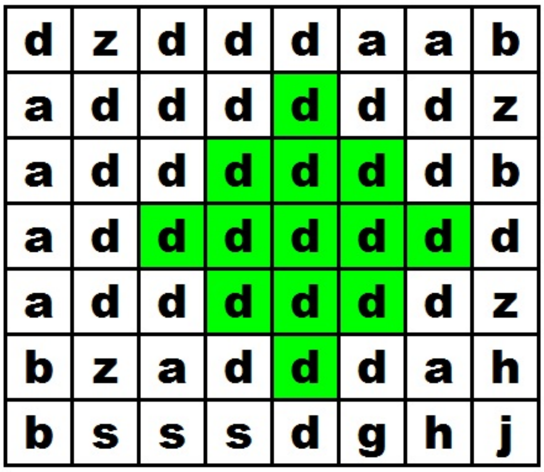
time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

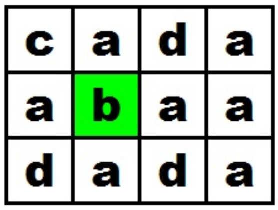
Carousel Boutique is busy again! Rarity has decided to visit the pony ball and she surely needs a new dress, because going out in the same dress several times is a sign of bad manners. First of all, she needs a dress pattern, which she is going to cut out from the rectangular piece of the multicolored fabric.

The piece of the multicolored fabric consists of $n \times m$ separate square scraps. Since Rarity likes dresses in style, a dress pattern must only include scraps sharing the same color. A dress pattern must be the square, and since Rarity is fond of rhombuses, the sides of a pattern must form a 45° angle with sides of a piece of fabric (that way it will be resembling the traditional picture of a rhombus).

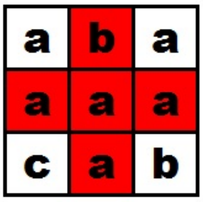
Examples of proper dress patterns:

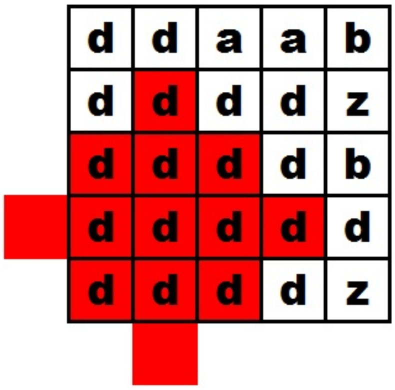


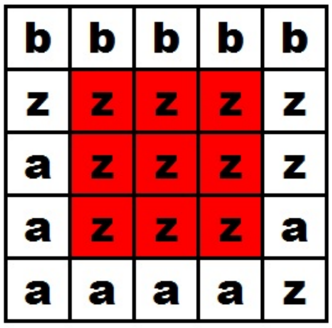




Examples of improper dress patterns:







The first one consists of multi-colored scraps, the second one goes beyond the bounds of the piece of fabric, the third one is not a square with sides forming a 45° angle with sides of the piece of fabric.

Rarity wonders how many ways to cut out a dress pattern that satisfies all the conditions that do exist. Please help her and satisfy her curiosity so she can continue working on her new masterpiece!

Input

The first line contains two integers n and m ($1 \leq n, m \leq 2000$). Each of the next n lines contains m characters: lowercase English letters, the j-th of which corresponds to scrap in the current line and in the j-th column. Scraps having the same letter share the same color, scraps having different letters have different colors.

Output

Print a single integer: the number of ways to cut out a dress pattern to satisfy all of Rarity's conditions.

Examples

input

```
3 3
aaa
aaa
aaa
```

output

```
10
```

input

```
3 4
abab
baba
abab
```

output

```
12
```

input

```
5 5
zbacg
baaac
aaaaa
eaaad
weadd
```

output

```
31
```

Note

In the first example, all the dress patterns of size 1 and one of size 2 are satisfactory.

In the second example, only the dress patterns of size 1 are satisfactory.

E1. Twilight and Ancient Scroll (easier version)

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an easier version of the problem E with smaller constraints.

Twilight Sparkle has received a new task from Princess Celestia. This time she asked to decipher the ancient scroll containing important knowledge of pony origin.

To hide the crucial information from evil eyes, pony elders cast a spell on the scroll. That spell adds exactly one letter in any place to each word it is cast on. To make the path to the knowledge more tangled elders chose *some* of words in the scroll and cast a spell on them.

Twilight Sparkle knows that the elders admired the order in all things so the scroll original scroll contained words in **lexicographically non-decreasing order**. She is asked to delete one letter from some of the words of the scroll (to undo the spell) to get some version of the original scroll.

Unfortunately, there may be more than one way to recover the ancient scroll. To not let the important knowledge slip by Twilight has to look through all variants of the original scroll and find the required one. To estimate the maximum time Twilight may spend on the work she needs to know the number of variants she has to look through. She asks you to find that number! Since that number can be very big, Twilight asks you to find it modulo $10^9 + 7$.

It may occur that princess Celestia has sent a wrong scroll so the answer may not exist.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

The first line contains a single integer n ($1 \le n \le 1000$): the number of words in the scroll.

The i-th of the next n lines contains a string consisting of lowercase English letters: the i-th word in the scroll. The length of each word is more or equal than 1. The sum of lengths of words does not exceed 20000.

Output

Print one integer: the number of ways to get a version of the original from the scroll modulo $10^9 + 7$.

Examples

input

```
3
abcd
zaza
ataka
```

output

```
4
```

input

```
4
dfs
bfs
sms
mms
```

output

```
8
```

input

```
3
abc
bcd
a
```

output

```
0
```

input

```
6
lapochka
kartyshka
bigbabytape
morgenshtern
ssshhhiiittt
```

```
queen
```

output

```
2028
```

Note

Notice that the elders could have written an empty word (but they surely cast a spell on it so it holds a length 1 now).

E2. Twilight and Ancient Scroll (harder version)

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is a harder version of the problem E with larger constraints.

Twilight Sparkle has received a new task from Princess Celestia. This time she asked to decipher the ancient scroll containing important knowledge of pony origin.

To hide the crucial information from evil eyes, pony elders cast a spell on the scroll. That spell adds exactly one letter in any place to each word it is cast on. To make the path to the knowledge more tangled elders chose *some* of words in the scroll and cast a spell on them.

Twilight Sparkle knows that the elders admired the order in all things so the scroll original scroll contained words in **lexicographically non-decreasing order**. She is asked to delete one letter from some of the words of the scroll (to undo the spell) to get some version of the original scroll.

Unfortunately, there may be more than one way to recover the ancient scroll. To not let the important knowledge slip by Twilight has to look through all variants of the original scroll and find the required one. To estimate the maximum time Twilight may spend on the work she needs to know the number of variants she has to look through. She asks you to find that number! Since that number can be very big, Twilight asks you to find it modulo 10^9+7.

It may occur that princess Celestia has sent a wrong scroll so the answer may not exist.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

The first line contains a single integer n ($1 \le n \le 10^5$): the number of words in the scroll.

The i-th of the next n lines contains a string consisting of lowercase English letters: the i-th word in the scroll. The length of each word is at least one. The sum of lengths of words does not exceed 10^6.

Output

Print one integer: the number of ways to get a version of the original from the scroll modulo 10^9+7.

Examples

input

```
3
abcd
zaza
ataka
```

output

```
4
```

input

```
4
dfs
bfs
sms
mms
```

output

```
8
```

input

```
3
abc
bcd
a
```

output

```
0
```

input

```
6
lapochka
kartyshka
bigbabytape
morgenshtern
ssshhhiiittt
queen
```

output

```
2028
```

Note

Notice that the elders could have written an empty word (but they surely cast a spell on it so it holds a length 1 now).