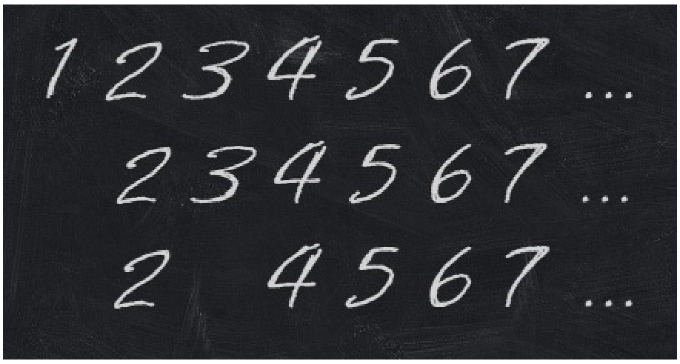
Educational Codeforces Round 68 (Rated for Div. 2)

A. Remove a Progression

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

You have a list of numbers from 1 to n written from left to right on the blackboard.

You perform an algorithm consisting of several steps (steps are 1-indexed). On the i-th step you wipe the i-th number (considering only **remaining** numbers). You wipe *the whole number* (not one digit).



When there are less than i numbers remaining, you stop your algorithm.

Now you wonder: what is the value of the x-th remaining number after the algorithm is stopped?

Input

The first line contains one integer T ($1 \le T \le 100$) — the number of queries. The next T lines contain queries — one per line. All queries are independent.

Each line contains two space-separated integers n and x ($1 \le x < n \le 10^9$) — the length of the list and the position we wonder about. It's guaranteed that after the algorithm ends, the list will still contain at least x numbers.

Output

Print T integers (one per query) — the values of the x-th number after performing the algorithm for the corresponding queries.

Example

input

```
3
3 1
4 2
69 6
```

output

```
2
4
12
```

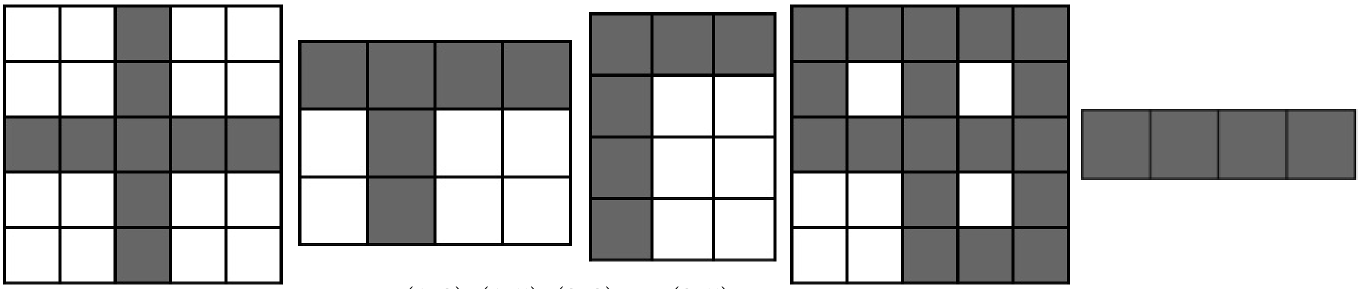
B. Yet Another Crosses Problem

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

You are given a picture consisting of n rows and m columns. Rows are numbered from 1 to n from the top to the bottom, columns are numbered from 1 to m from the left to the right. Each cell is painted either black or white.

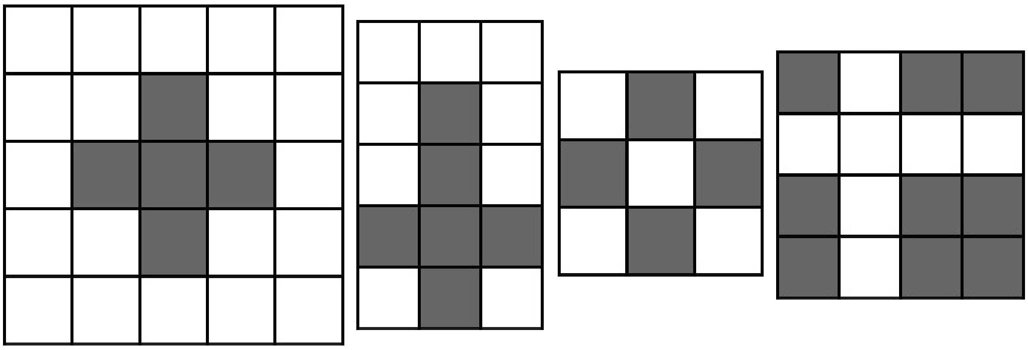
You think that this picture is not interesting enough. You consider a picture to be interesting if there is at least one *cross* in it. A cross is represented by a pair of numbers x and y, where $1 \le x \le n$ and $1 \le y \le m$, such that **all cells** in row x and **all cells** in column y are painted black.

For examples, each of these pictures contain crosses:



The fourth picture contains 4 crosses: at $(1, 3)$, $(1, 5)$, $(3, 3)$ and $(3, 5)$.

Following images don't contain crosses:



You have a brush and a can of black paint, so you can make this picture interesting. Each minute you may choose a white cell and paint it black.

What is the minimum number of minutes you have to spend so the resulting picture contains at least one cross?

You are also asked to answer multiple independent queries.

Input

The first line contains an integer q ($1 \le q \le 5 \cdot 10^4$) — the number of queries.

The first line of each query contains two integers n and m ($1 \le n, m \le 5 \cdot 10^4$, $n \cdot m \le 4 \cdot 10^5$) — the number of rows and the number of columns in the picture.

Each of the next n lines contains m characters — '.' if the cell is painted white and '*' if the cell is painted black.

It is guaranteed that $\sum n \le 5 \cdot 10^4$ and $\sum n \cdot m \le 4 \cdot 10^5$.

Output

Print q lines, the i-th line should contain a single integer — the answer to the i-th query, which is the minimum number of minutes you have to spend so the resulting picture contains at least one cross.

Example

input

```
9
5 5
..*..
..*..
*****
..*..
..*..
3 4
****
.*..
.*..
4 3
***
*..
*..
*..
5 5
*****
*.*.*
*****
..*.*
..***
1 4
****
5 5
.....
..*..
.***.
..*..
.....
5 3
...
.*.
.*.
```

```
***
.*.
3 3
.*.
*.*
.*.
4 4
*.**
....
*.**
*.**
```

output

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

Note

The example contains all the pictures from above in the same order.

The first 5 pictures already contain a cross, thus you don't have to paint anything.

You can paint $(1, 3)$, $(3, 1)$, $(5, 3)$ and $(3, 5)$ on the 6-th picture to get a cross in $(3, 3)$. That'll take you 4 minutes.

You can paint $(1, 2)$ on the 7-th picture to get a cross in $(4, 2)$.

You can paint $(2, 2)$ on the 8-th picture to get a cross in $(2, 2)$. You can, for example, paint $(1, 3)$, $(3, 1)$ and $(3, 3)$ to get a cross in $(3, 3)$ but that will take you 3 minutes instead of 1.

There are 9 possible crosses you can get in minimum time on the 9-th picture. One of them is in $(1, 1)$: paint $(1, 2)$ and $(2, 1)$.

C. From S To T

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given three strings s, t and p consisting of lowercase Latin letters. You may perform any number (possibly, zero) operations on these strings.

During each operation you choose any character from p, erase it from p and insert it into string s (you may insert this character anywhere you want: in the beginning of s, in the end or between any two consecutive characters).

For example, if p is aba, and s is de, then the following outcomes are possible (the character we erase from p and insert into s is highlighted):

- **a**ba $\rightarrow$ ba, de $\rightarrow$ **a**de;
- **a**ba $\rightarrow$ ba, de $\rightarrow$ d**a**e;
- **a**ba $\rightarrow$ ba, de $\rightarrow$ de**a**;
- a**b**a $\rightarrow$ aa, de $\rightarrow$ **b**de;
- a**b**a $\rightarrow$ aa, de $\rightarrow$ d**b**e;
- a**b**a $\rightarrow$ aa, de $\rightarrow$ de**b**;
- ab**a** $\rightarrow$ ab, de $\rightarrow$ **a**de;
- ab**a** $\rightarrow$ ab, de $\rightarrow$ d**a**e;
- ab**a** $\rightarrow$ ab, de $\rightarrow$ de**a**;

Your goal is to perform several (maybe zero) operations so that s becomes equal to t. Please determine whether it is possible.

Note that you have to answer q independent queries.

Input

The first line contains one integer q $(1 \le q \le 100)$ — the number of queries. Each query is represented by three consecutive lines.

The first line of each query contains the string s $(1 \le |s| \le 100)$ consisting of lowercase Latin letters.

The second line of each query contains the string t $(1 \le |t| \le 100)$ consisting of lowercase Latin letters.

The third line of each query contains the string p $(1 \le |p| \le 100)$ consisting of lowercase Latin letters.

Output

For each query print YES if it is possible to make s equal to t, and NO otherwise.

You may print every letter in any case you want (so, for example, the strings yEs, yes, Yes and YES will all be recognized as positive answer).

Example

input

```
4
ab
acxb
cax
a
aaaa
aaabbcc
a
aaaa
aabbcc
ab
baaa
aaaaa
```

output

```
YES
YES
NO
NO
```

Note

In the first test case there is the following sequence of operation:

1. $s =$ ab, $t =$ acxb, $p =$ cax;
2. $s =$ acb, $t =$ acxb, $p =$ ax;
3. $s =$ acxb, $t =$ acxb, $p =$ a.

In the second test case there is the following sequence of operation:

1. $s =$ a, $t =$ aaaa, $p =$ aaabbcc;
2. $s =$ aa, $t =$ aaaa, $p =$ aabbcc;
3. $s =$ aaa, $t =$ aaaa, $p =$ abbcc;
4. $s =$ aaaa, $t =$ aaaa, $p =$ bbcc.

D. 1-2-K Game

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

Alice and Bob play a game. There is a paper strip which is divided into $n + 1$ cells numbered from left to right starting from 0. There is a chip placed in the n-th cell (the last one).

Players take turns, Alice is first. Each player during his or her turn has to move the chip 1, 2 or k cells to the left (so, if the chip is currently in the cell i, the player can move it into cell $i - 1$, $i - 2$ or $i - k$). The chip should not leave the borders of the paper strip: it is impossible, for example, to move it k cells to the left if the current cell has number $i < k$. The player who can't make a move loses the game.

Who wins if both participants play optimally?

Alice and Bob would like to play several games, so you should determine the winner in each game.

Input

The first line contains the single integer T ($1 \le T \le 100$) — the number of games. Next T lines contain one game per line. All games are independent.

Each of the next T lines contains two integers n and k ($0 \le n \le 10^9$, $3 \le k \le 10^9$) — the length of the strip and the constant denoting the third move, respectively.

Output

For each game, print `Alice` if Alice wins this game and `Bob` otherwise.

Example

input

```
4
0 3
3 3
3 4
4 4
```

output

```
Bob
Alice
Bob
Alice
```

E. Count The Rectangles

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

There are n segments drawn on a plane; the i-th segment connects two points ($x_{i,1}$, $y_{i,1}$) and ($x_{i,2}$, $y_{i,2}$). Each segment is non-degenerate, and is either horizontal or vertical — formally, for every $i \in [1, n]$ either $x_{i,1} = x_{i,2}$ or $y_{i,1} = y_{i,2}$ (but only one of these conditions holds). Only segments of different types may intersect: no pair of horizontal segments shares any common points, and no pair of vertical segments shares any common points.

We say that four segments having indices h_1, h_2, v_1 and v_2 such that $h_1 < h_2$ and $v_1 < v_2$ form a rectangle if the following conditions hold:

- segments h_1 and h_2 are horizontal;
- segments v_1 and v_2 are vertical;
- segment h_1 intersects with segment v_1;
- segment h_2 intersects with segment v_1;
- segment h_1 intersects with segment v_2;
- segment h_2 intersects with segment v_2.

Please calculate the number of ways to choose four segments so they form a rectangle. Note that the conditions $h_1 < h_2$ and $v_1 < v_2$ should hold.

Input

The first line contains one integer n ($1 \leq n \leq 5000$) — the number of segments.

Then n lines follow. The i-th line contains four integers $x_{i,1}$, $y_{i,1}$, $x_{i,2}$ and $y_{i,2}$ denoting the endpoints of the i-th segment. All coordinates of the endpoints are in the range $[-5000, 5000]$.

It is guaranteed that each segment is non-degenerate and is either horizontal or vertical. Furthermore, if two segments share a common point, one of these segments is horizontal, and another one is vertical.

Output

Print one integer — the number of ways to choose four segments so they form a rectangle.

Examples

input

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

output

```
7
```

input

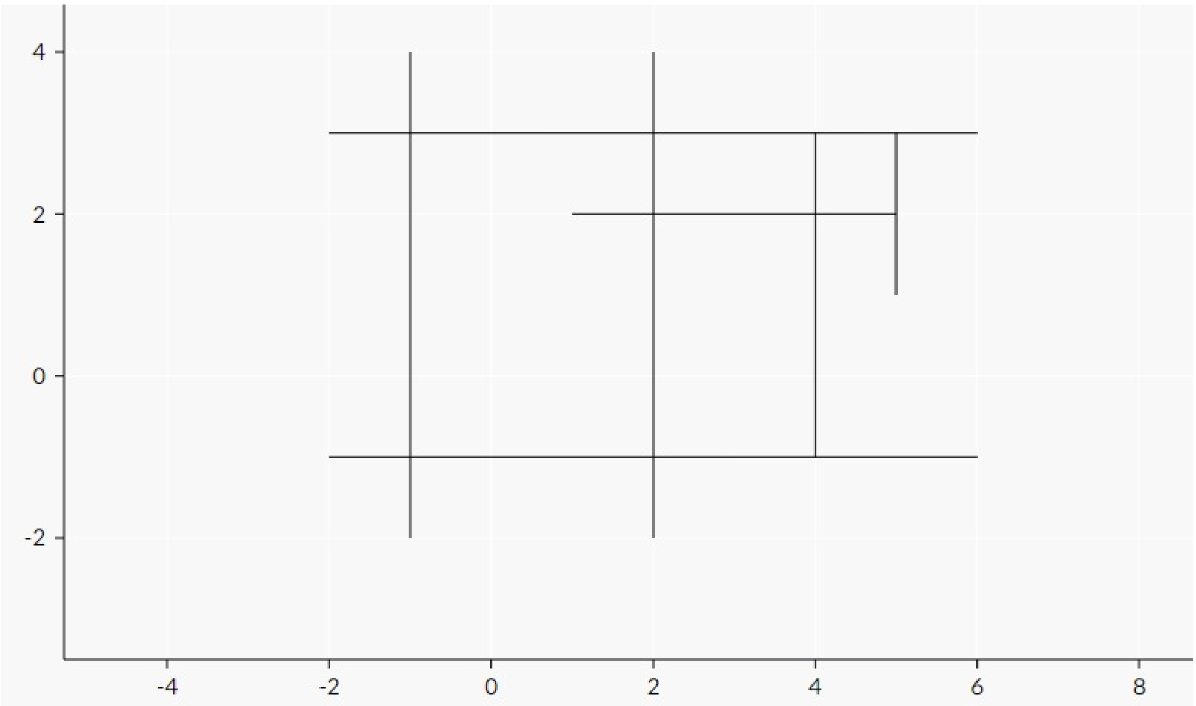
```
5
1 5 1 0
0 1 5 1
5 4 0 4
4 2 4 0
4 3 4 5
```

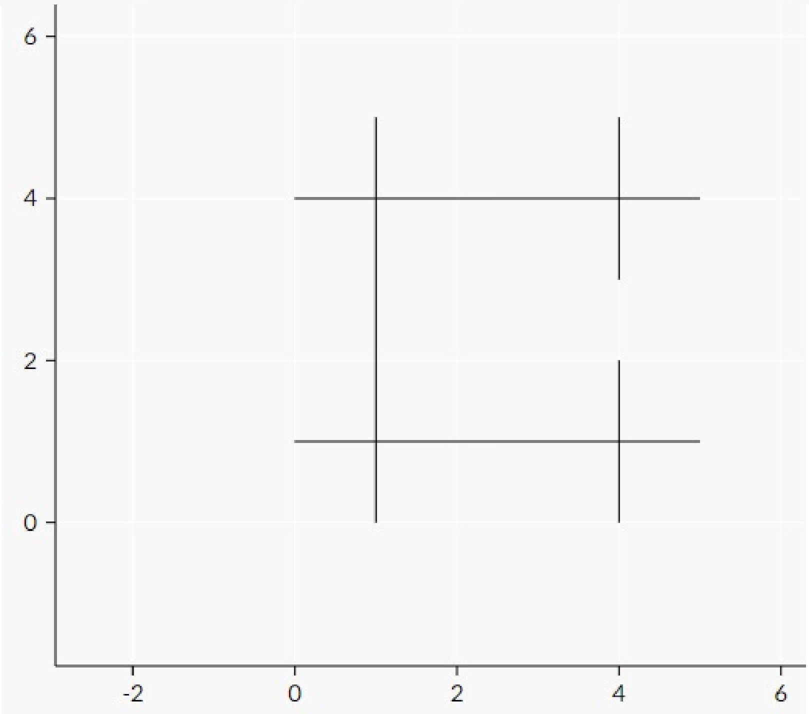
output

```
0
```

Note

The following pictures represent sample cases:





F. Crossword Expert

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

Today Adilbek is taking his probability theory test. Unfortunately, when Adilbek arrived at the university, there had already been a long queue of students wanting to take the same test. Adilbek has estimated that he will be able to start the test only T seconds after coming.

Fortunately, Adilbek can spend time without revising any boring theorems or formulas. He has an app on this smartphone which contains n Japanese crosswords to solve. Adilbek has decided to solve them all one by one in the order they are listed in the app, without skipping any crossword. For each crossword, a number t_i is given that represents the time it takes an average crossword expert to solve this crossword (the time is given in seconds).

Adilbek is a true crossword expert, but, unfortunately, he is sometimes unlucky in choosing the way to solve the crossword. So, it takes him either t_i seconds or $t_i + 1$ seconds to solve the i-th crossword, equiprobably (with probability $\frac{1}{2}$ he solves the crossword in exactly t_i seconds, and with probability $\frac{1}{2}$ he has to spend an additional second to finish the crossword). All these events are independent.

After T seconds pass (or after solving the last crossword, if he manages to do it in less than T seconds), Adilbek closes the app (if he finishes some crossword at the same moment, that crossword is considered solved; otherwise Adilbek does not finish solving the current crossword at all). He thinks it would be an interesting probability theory problem to calculate E — the expected number of crosswords he will be able to solve completely. Can you calculate it?

Recall that the expected value of a discrete random variable is the probability-weighted average of all possible values — in this problem it means that the expected value of the number of solved crosswords can be calculated as $E = \sum_{i=0}^{n} i p_i$, where p_i is the probability that Adilbek will solve exactly i crosswords.

We can represent E as rational fraction $\frac{P}{Q}$ with $Q > 0$. To give the answer, you should print $P \cdot Q^{-1} \bmod (10^9 + 7)$.

Input

The first line contains two integers n and T ($1 \le n \le 2 \cdot 10^5$, $1 \le T \le 2 \cdot 10^{14}$) — the number of crosswords and the time Adilbek has to spend, respectively.

The second line contains n integers $t_1, t_2, \dots, t_n$ ($1 \le t_i \le 10^9$), where t_i is the time it takes a crossword expert to solve the i-th crossword.

Note that Adilbek solves the crosswords in the order they are given in the input without skipping any of them.

Output

Print one integer — the expected value of the number of crosswords Adilbek solves in T seconds, expressed in the form of $P \cdot Q^{-1} \bmod (10^9 + 7)$.

Examples

input

```
3 5
2 2 2
```

output

```
750000007
```

input

```
3 5
2 1 2
```

output

```
125000003
```

Note

The answer for the first sample is equal to $\frac{14}{8}$.

The answer for the second sample is equal to $\frac{17}{8}$.

G. Another Meme Problem

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Let's call a fraction $\frac{x}{y}$ good if there exists at least one another fraction $\frac{x'}{y'}$ such that $\frac{x}{y} = \frac{x'}{y'}$, $1 \le x', y' \le 9$, the digit denoting x' is contained in the decimal representation of x, and the digit denoting y' is contained in the decimal representation of y. For example, $\frac{26}{13}$ is a good fraction, because $\frac{26}{13} = \frac{2}{1}$.

You are given an integer number n. Please calculate the number of good fractions $\frac{x}{y}$ such that $1 \le x \le n$ and $1 \le y \le n$. The answer may be really large, so print it modulo 998244353.

Input

The only line of the input contains one integer n ($1 \le n < 10^{100}$).

Output

Print the number of good fractions $\frac{x}{y}$ such that $1 \le x \le n$ and $1 \le y \le n$. The answer may be really large, so print it modulo 998244353.

Examples

input

```
42
```

output

```
150
```

input

```
3141592653589793238462643383279
```

output

```
459925407
```