Codeforces Round #706 (Div. 2)

A. Split it!

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

Kawashiro Nitori is a girl who loves competitive programming.

One day she found a string and an integer. As an advanced problem setter, she quickly thought of a problem.

Given a string s and a parameter k, you need to check if there exist $k+1$ non-empty strings $a_1, a_2 \ldots, a_{k+1}$, such that

$$s = a_1 + a_2 + \ldots + a_k + a_{k+1} + R(a_k) + R(a_{k-1}) + \ldots + R(a_1).$$

Here $+$ represents concatenation. We define $R(x)$ as a reversed string x. For example $R(abcd) = dcba$. Note that in the formula above the part $R(a_{k+1})$ is intentionally skipped.

Input

The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case description contains two integers n, k ($1 \le n \le 100$, $0 \le k \le \lfloor \frac{n}{2} \rfloor$) — the length of the string s and the parameter k.

The second line of each test case description contains a single string s of length n, consisting of lowercase English letters.

Output

For each test case, print "YES" (without quotes), if it is possible to find $a_1, a_2, \ldots, a_{k+1}$, and "NO" (without quotes) otherwise.

You can print letters in any case (upper or lower).

Example

input
7 5 1 qwqwq 2 1 ab 3 1 ioi 4 2 icpc 22 0 dokidokiliteratureclub 19 8 imteamshanghaialice 6 3 aaaaaa

output
YES NO YES NO YES NO NO

Note

In the first test case, one possible solution is $a_1 = qw$ and $a_2 = q$.

In the third test case, one possible solution is $a_1 = i$ and $a_2 = o$.

In the fifth test case, one possible solution is $a_1 = dokidokiliteratureclub$.

B. Max and Mex

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

You are given a multiset S initially consisting of n distinct non-negative integers. A multiset is a set, that can contain some elements multiple times.

You will perform the following operation k times:

- Add the element $\lceil \frac{a+b}{2} \rceil$ (rounded up) into S, where $a = \operatorname{mex}(S)$ and $b = \max(S)$. If this number is already in the set, it is added again.

Here $\max$ of a multiset denotes the maximum integer in the multiset, and mex of a multiset denotes the smallest non-negative integer that is not present in the multiset. For example:

- $\operatorname{mex}(\{1,4,0,2\}) = 3$;
- $\operatorname{mex}(\{2,5,1\}) = 0$.

Your task is to calculate the number of **distinct** elements in S after k operations will be done.

Input
The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n, k ($1 \le n \le 10^5$, $0 \le k \le 10^9$) — the initial size of the multiset S and how many operations you need to perform.

The second line of each test case contains n **distinct** integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — the numbers in the initial multiset.

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

Output
For each test case, print the number of **distinct** elements in S after k operations will be done.

Example

input
5 4 1 0 1 3 4 3 1 0 1 4 3 0 0 1 4 3 2 0 1 2 3 2 1 2 3
output
4 4 3 5 3

Note
In the first test case, $S = \{0,1,3,4\}$, $a = \operatorname{mex}(S) = 2$, $b = \max(S) = 4$, $\lceil \frac{a+b}{2} \rceil = 3$. So 3 is added into S, and S becomes $\{0,1,3,3,4\}$. The answer is 4.

In the second test case, $S = \{0,1,4\}$, $a = \operatorname{mex}(S) = 2$, $b = \max(S) = 4$, $\lceil \frac{a+b}{2} \rceil = 3$. So 3 is added into S, and S becomes $\{0,1,3,4\}$. The answer is 4.

C. Diamond Miner

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

Diamond Miner is a game that is similar to Gold Miner, but there are n miners instead of 1 in this game.

The mining area can be described as a plane. The n miners can be regarded as n points **on the y-axis**. There are n diamond mines in the mining area. We can regard them as n points **on the x-axis**. For some reason, **no miners or diamond mines can be at the origin** (point $(0,0)$).

Every miner should mine **exactly** one diamond mine. Every miner has a hook, which can be used to mine a diamond mine. If a miner at the point (a,b) uses his hook to mine a diamond mine at the point (c,d), he will spend $\sqrt{(a-c)^2+(b-d)^2}$ energy to mine it (the distance between these points). The miners can't move or help each other.

The object of this game is to minimize **the sum of the energy** that miners spend. Can you find this minimum?

Input
The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the number of miners and mines.

Each of the next $2n$ lines contains two space-separated integers x ($-10^8 \le x \le 10^8$) and y ($-10^8 \le y \le 10^8$), which represent the point (x,y) to describe **a miner's or a diamond mine's** position. Either $x = 0$, meaning there is a miner at the point $(0,y)$, or $y = 0$, meaning there is a diamond mine at the point $(x,0)$. There can be multiple miners or diamond mines at the same point.

It is guaranteed that no point is at the origin. It is guaranteed that the number of points on the x-axis is equal to n and the number of points on the y-axis is equal to n.

It's guaranteed that the sum of n for all test cases does not exceed 10^5.

Output
For each test case, print a single real number — the minimal sum of energy that should be spent.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-9}.

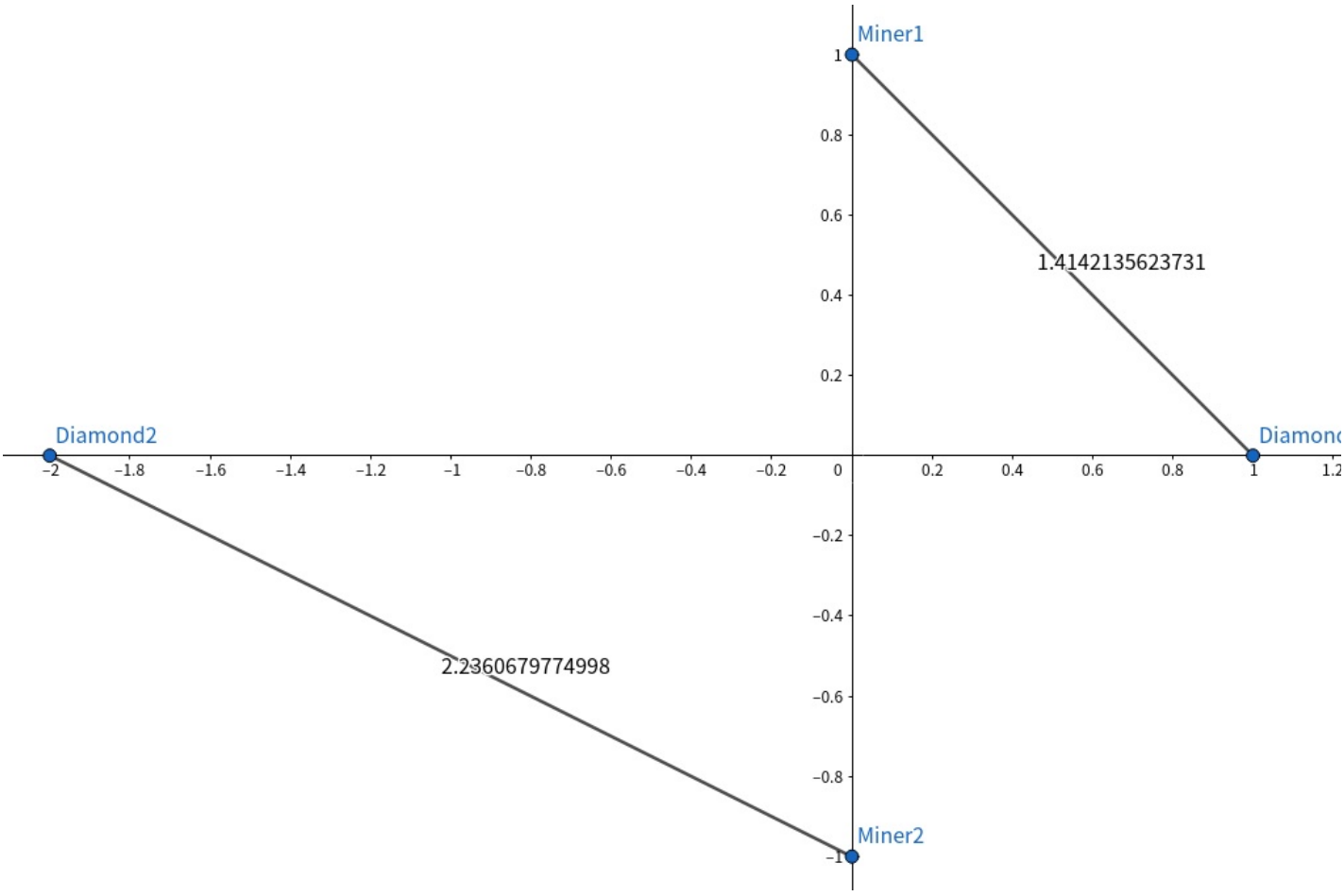
Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-9}$.

Example

input
3 2 0 1 1 0 0 -1 -2 0 4 1 0 3 0 -5 0 6 0 0 3 0 1 0 2 0 4 5 3 0 0 4 0 -3 4 0 2 0 1 0 -3 0 0 -10 0 -2 0 -10
output
3.650281539872885 18.061819283610362 32.052255376143336

Note

In the first test case, the miners are at $(0,1)$ and $(0,-1)$, while the diamond mines are at $(1,0)$ and $(-2,0)$. If you arrange the miners to get the diamond mines in the way, shown in the picture, you can get the sum of the energy $\sqrt{2}+\sqrt{5}$.



D. Let's Go Hiking

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

On a weekend, Qingshan suggests that she and her friend Daniel go hiking. Unfortunately, they are busy high school students, so they can only go hiking on scratch paper.

A permutation p is written from left to right on the paper. First Qingshan chooses an integer index x ($1 \le x \le n$) and tells it to Daniel. After that, Daniel chooses another integer index y ($1 \le y \le n$, $y \ne x$).

The game progresses turn by turn and as usual, Qingshan moves first. The rules follow:

- If it is Qingshan's turn, Qingshan must change x to such an index x' that $1 \le x' \le n$, $|x' - x| = 1$, $x' \ne y$, and $p_{x'} < p_x$ at the same time.
- If it is Daniel's turn, Daniel must change y to such an index y' that $1 \le y' \le n$, $|y' - y| = 1$, $y' \ne x$, and $p_{y'} > p_y$ at the same time.

The person who can't make her or his move loses, and the other wins. You, as Qingshan's fan, are asked to calculate the number of possible x to make Qingshan win in the case both players play optimally.

Input
The first line contains a single integer n ($2 \le n \le 10^5$) — the length of the permutation.

The second line contains n distinct integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$) — the permutation.

Output
Print the number of possible values of x that Qingshan can choose to make her win.

Examples

input
5 1 2 5 4 3
output
1

input
7 1 2 4 6 5 3 7
output
0

Note
In the first test case, Qingshan can only choose $x = 3$ to win, so the answer is 1.

In the second test case, if Qingshan will choose $x = 4$, Daniel can choose $y = 1$. In the first turn (Qingshan's) Qingshan chooses $x' = 3$ and changes x to 3. In the second turn (Daniel's) Daniel chooses $y' = 2$ and changes y to 2. Qingshan can't choose $x' = 2$ because $y = 2$ at this time. Then Qingshan loses.

E. Garden of the Sun

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

There are many sunflowers in the Garden of the Sun.

Garden of the Sun is a rectangular table with n rows and m columns, where the cells of the table are farmlands. All of the cells grow a sunflower on it. Unfortunately, one night, the lightning stroke some (possibly zero) cells, and sunflowers on those cells were burned into ashes. In other words, those cells struck by the lightning became empty. Magically, **any two empty cells have no common points** (neither edges nor corners).

Now the owner wants to remove some (possibly zero) sunflowers to reach the following two goals:

- When you are on an empty cell, you can walk to any other empty cell. In other words, those empty cells are connected.
- There is **exactly one** simple path between any two empty cells. In other words, there is no cycle among the empty cells.

You can walk from an empty cell to another if they share a common edge.

Could you please give the owner a solution that meets all her requirements?

Note that you are not allowed to plant sunflowers. You **don't need** to minimize the number of sunflowers you remove. It can be shown that the answer always exists.

Input
The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The description of the test cases follows.

The first line contains two integers n, m ($1 \le n, m \le 500$) — the number of rows and columns.

Each of the next n lines contains m characters. Each character is either 'X' or '`.`', representing an empty cell and a cell that grows a sunflower, respectively.

It is guaranteed that the sum of $n \cdot m$ for all test cases does not exceed $250\,000$.

Output
For each test case, print n lines. Each should contain m characters, representing one row of the table. Each character should be either 'X' or '`.`', representing an empty cell and a cell with a sunflower, respectively.

If there are multiple answers, you can print any. It can be shown that the answer always exists.

Example

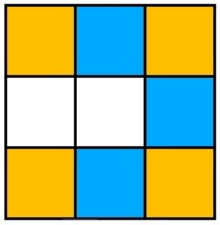
input
5 3 3 X.X ... X.X 4 4 .... .X.X .... .X.X 5 5 .X... ....X .X... ..... X.X.X 1 10 ....X.X.X. 2 2 .. ..

output
XXX ..X XXX XXXX .X.X .X.. .XXX .X... .XXXX .X... .X... XXXXX XXXXXXXXXX .. ..

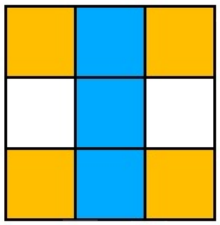
Note

Let's use (x, y) to describe the cell on x-th row and y-th column.

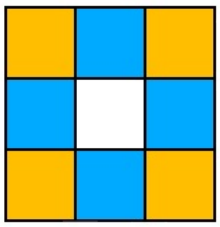
In the following pictures white, yellow, and blue cells stand for the cells that grow a sunflower, the cells lightning stroke, and the cells sunflower on which are removed, respectively.



In the first test case, one possible solution is to remove sunflowers on $(1, 2)$, $(2, 3)$ and $(3, 2)$.

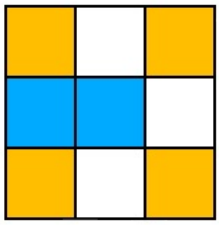


Another acceptable solution is to remove sunflowers on $(1, 2)$, $(2, 2)$ and $(3, 2)$.



This output is considered wrong because there are 2 simple paths between any pair of cells (there is a cycle). For example, there are 2 simple paths between $(1, 1)$ and $(3, 3)$.

1. $(1, 1) \to (1, 2) \to (1, 3) \to (2, 3) \to (3, 3)$
2. $(1, 1) \to (2, 1) \to (3, 1) \to (3, 2) \to (3, 3)$



This output is considered wrong because you can't walk from $(1, 1)$ to $(3, 3)$.

F. BFS Trees

time limit per test: 2.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

We define a spanning tree of a graph to be a BFS tree *rooted at* vertex s if and only if for every node t the shortest distance between s and t in the graph is equal to the shortest distance between s and t in the spanning tree.

Given a graph, we define $f(x, y)$ to be the number of spanning trees of that graph that are BFS trees rooted at vertices x and y at the same time.

You are given an undirected connected graph with n vertices and m edges. Calculate $f(i, j)$ for all i, j by modulo $998\,244\,353$.

Input

The first line contains two integers n, m ($1 \le n \le 400$, $0 \le m \le 600$) — the number of vertices and the number of edges in the graph.

The i-th of the next m lines contains two integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i < b_i$), representing an edge connecting a_i and b_i.

It is guaranteed that all edges are distinct and the graph is connected.

Output

Print n lines, each consisting of n integers.

The integer printed in the row i and the column j should be $f(i, j) \bmod 998\,244\,353$.

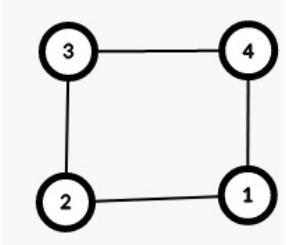
Examples

input
4 4 1 2 2 3 3 4 1 4
output
2 1 0 1 1 2 1 0 0 1 2 1 1 0 1 2

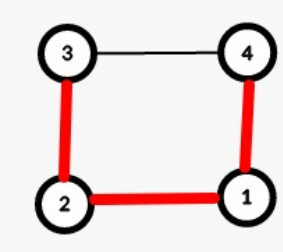
input
8 9 1 2 1 3 1 4 2 7 3 5 3 6 4 8 2 3 3 4
output
1 0 0 0 0 0 0 0 0 2 0 0 0 0 2 0 0 0 1 0 1 1 0 0 0 0 0 2 0 0 0 2 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 2 0 0 0 0 2 0 0 0 0 2 0 0 0 2

Note

The following picture describes the first example.



The tree with red edges is a BFS tree rooted at both 1 and 2.



Similarly, the BFS tree for other adjacent pairs of vertices can be generated in this way.

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform