A. AB Balance

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

You are given a string s of length n consisting of characters a and/or b.

Let $\mathrm{AB}(s)$ be the number of occurrences of string ab in s as a **substring**. Analogically, $\mathrm{BA}(s)$ is the number of occurrences of ba in s as a **substring**.

In one step, you can choose any index i and replace s_i with character a or b.

What is the minimum number of steps you need to make to achieve $\mathrm{AB}(s) = \mathrm{BA}(s)$?

Reminder:

The number of occurrences of string d in s as substring is the number of indices i ($1 \le i \le |s| - |d| + 1$) such that substring $s_i s_{i+1} \dots s_{i+|d|-1}$ is equal to d. For example, $\mathrm{AB}($aabbbabaa$) = 2$ since there are two indices i: $i = 2$ where a<u>ab</u>bbabaa and $i = 6$ where aabbb<u>ab</u>aa.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). Description of the test cases follows.

The first and only line of each test case contains a single string s ($1 \le |s| \le 100$, where $|s|$ is the length of the string s), consisting only of characters a and/or b.

Output

For each test case, print the resulting string s with $\mathrm{AB}(s) = \mathrm{BA}(s)$ you'll get making the minimum number of steps.

If there are multiple answers, print any of them.

Example

input

```
4
b
aabbbabaa
abbb
abbaab
```

output

```
b
aabbbabaa
bbbb
abbaaa
```

Note

In the first test case, both $\mathrm{AB}(s) = 0$ and $\mathrm{BA}(s) = 0$ (there are no occurrences of ab (ba) in b), so can leave s untouched.

In the second test case, $\mathrm{AB}(s) = 2$ and $\mathrm{BA}(s) = 2$, so you can leave s untouched.

In the third test case, $\mathrm{AB}(s) = 1$ and $\mathrm{BA}(s) = 0$. For example, we can change s_1 to b and make both values zero.

In the fourth test case, $\mathrm{AB}(s) = 2$ and $\mathrm{BA}(s) = 1$. For example, we can change s_6 to a and make both values equal to 1.

B. Update Files

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

Berland State University has received a new update for the operating system. Initially it is installed only on the 1-st computer.

Update files should be copied to all n computers. The computers are not connected to the internet, so the only way to transfer update files from one computer to another is to copy them using a patch cable (a cable connecting two computers directly). Only one patch cable can be connected to a computer at a time. Thus, from any computer where the update files are installed, they can be copied to some other computer in exactly one hour.

Your task is to find the minimum number of hours required to copy the update files to all n computers if there are only k patch cables in Berland State University.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

Each test case consists of a single line that contains two integers n and k ($1 \le k \le n \le 10^{18}$) — the number of computers and the number of patch cables.

Output

For each test case print one integer — the minimum number of hours required to copy the update files to all n computers.

Example

input

```
4
8 3
6 6
7 1
1 1
```

output

```
4
3
6
0
```

Note

Let's consider the test cases of the example:

- $n = 8$, $k = 3$:
  1. during the first hour, we copy the update files from the computer 1 to the computer 2;
  2. during the second hour, we copy the update files from the computer 1 to the computer 3, and from the computer 2 to the computer 4;
  3. during the third hour, we copy the update files from the computer 1 to the computer 5, from the computer 2 to the computer 6, and from the computer 3 to the computer 7;
  4. during the fourth hour, we copy the update files from the computer 2 to the computer 8.
- $n = 6$, $k = 6$:
  1. during the first hour, we copy the update files from the computer 1 to the computer 2;
  2. during the second hour, we copy the update files from the computer 1 to the computer 3, and from the computer 2 to the computer 4;
  3. during the third hour, we copy the update files from the computer 1 to the computer 5, and from the computer 2 to the computer 6.
- $n = 7$, $k = 1$:
  1. during the first hour, we copy the update files from the computer 1 to the computer 2;
  2. during the second hour, we copy the update files from the computer 1 to the computer 3;
  3. during the third hour, we copy the update files from the computer 1 to the computer 4;
  4. during the fourth hour, we copy the update files from the computer 4 to the computer 5;
  5. during the fifth hour, we copy the update files from the computer 4 to the computer 6;
  6. during the sixth hour, we copy the update files from the computer 3 to the computer 7.

C. Banknotes

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

In Berland, n different types of banknotes are used. Banknotes of the i-th type have denomination 10^{a_i} burles (burles are the currency used in Berland); the denomination of banknotes of the first type is exactly 1.

Let's denote $f(s)$ as the minimum number of banknotes required to represent exactly s burles. For example, if the denominations of banknotes used in Berland are 1, 10 and 100, then $f(59) = 14$: 9 banknotes with denomination of 1 burle and 5 banknotes with denomination of 10 burles can be used to represent exactly $9 \cdot 1 + 5 \cdot 10 = 59$ burles, and there's no way to do it with fewer banknotes.

For a given integer k, find the minimum positive number of burles s that cannot be represented with k or fewer banknotes (that is, $f(s) > k$).

Input

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

The first line of each test case contains two integers n and k ($1 \le n \le 10; 1 \le k \le 10^9$).

The next line contains n integers $a_1, a_2, \dots, a_n$ ($0 = a_1 < a_2 < \dots < a_n \le 9$).

Output

For each test case, print one integer — the minimum positive number of burles s that cannot be represented with k or fewer banknotes.

Example

input

```
4
3 13
0 1 2
2 777
0 4
3 255
0 1 3
10 1000000000
0 1 2 3 4 5 6 7 8 9
```

output

```
59
778
148999
999999920999999999
```

D. Red-Blue Matrix

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a matrix, consisting of n rows and m columns. The j-th cell of the i-th row contains an integer a_{ij}.

First, you have to color each row of the matrix either red or blue in such a way that **at least one row is colored red** and **at least one row is colored blue**.

Then, you have to choose an integer k ($1 \le k < m$) and cut the colored matrix in such a way that the first k columns become a separate matrix (the *left* matrix) and the last $m - k$ columns become a separate matrix (the *right* matrix).

The coloring and the cut are called *perfect* if two properties hold:

- every red cell in the left matrix contains an integer greater than every blue cell in the left matrix;
- every blue cell in the right matrix contains an integer greater than every red cell in the right matrix.

Find any perfect coloring and cut, or report that there are none.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of testcases.

Then the descriptions of t testcases follow.

The first line of each testcase contains two integers n and m ($2 \le n, m \le 5 \cdot 10^5$; $n \cdot m \le 10^6$) — the number of rows and the number of columns in the matrix, respectively.

The i-th of the next n lines contains m integers $a_{i1}, a_{i2}, \dots, a_{im}$ ($1 \le a_{ij} \le 10^6$).

The sum of $n \cdot m$ over all testcases doesn't exceed 10^6.

Output

For each testcase print an answer. If there are no perfect colorings and cuts in the matrix, then print "NO".

Otherwise, first, print "YES". Then a string, consisting of n characters: the i-th character should be 'R' if the i-th row is colored red and 'B' if it's colored blue. The string should contain at least one 'R' and at least one 'B'. Finally, print an integer k ($1 \le k < m$) — the number of columns from the left that are cut.

Example

input

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

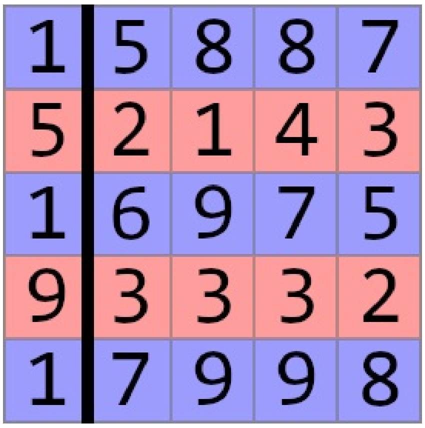
```
2 6
3 3 3 2 2 2
1 1 1 4 4 4
```

output

```
YES
BRBRB 1
NO
YES
RB 3
```

Note

The coloring and the cut for the first testcase:



E. Arena

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n heroes fighting in the arena. Initially, the i-th hero has a_i health points.

The fight in the arena takes place in several rounds. At the beginning of each round, each alive hero deals 1 damage to all other heroes. Hits of all heroes occur simultaneously. Heroes whose health is less than 1 at the end of the round are considered killed.

If exactly 1 hero remains alive after a certain round, then he is declared the winner. Otherwise, there is no winner.

Your task is to calculate the number of ways to choose the initial health points for each hero a_i, where $1 \le a_i \le x$, so that there is no winner of the fight. The number of ways can be very large, so print it modulo 998244353. Two ways are considered different if at least one hero has a different amount of health. For example, $[1, 2, 1]$ and $[2, 1, 1]$ are different.

Input

The only line contains two integers n and x ($2 \le n \le 500; 1 \le x \le 500$).

Output

Print one integer — the number of ways to choose the initial health points for each hero a_i, where $1 \le a_i \le x$, so that there is no winner of the fight, taken modulo 998244353.

Examples

input

```
2 5
```

output

```
5
```

input

```
3 3
```

output

```
15
```

input

```
5 4
```

output

```
1024
```

input

```
13 37
```

output
976890680

F. Tree Queries

time limit per test: 6 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

You are given a tree consisting of n vertices. Recall that a tree is an undirected connected acyclic graph. The given tree is rooted at the vertex 1.

You have to process q queries. In each query, you are given a vertex of the tree v and an integer k.

To process a query, you may delete any vertices from the tree in any order, except for the root and the vertex v. When a vertex is deleted, its children become the children of its parent. You have to process a query in such a way that maximizes the value of $c(v) - m \cdot k$ (where $c(v)$ is the resulting number of children of the vertex v, and m is the number of vertices you have deleted). Print the maximum possible value you can obtain.

The queries are independent: the changes you make to the tree while processing a query don't affect the tree in other queries.

Input

The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of vertices in the tree.

Then $n-1$ lines follow, the i-th of them contains two integers x_i and y_i ($1 \le x_i, y_i \le n$; $x_i \ne y_i$) — the endpoints of the i-th edge. These edges form a tree.

The next line contains one integer q ($1 \le q \le 2 \cdot 10^5$) — the number of queries.

Then q lines follow, the j-th of them contains two integers v_j and k_j ($1 \le v_j \le n$; $0 \le k_j \le 2 \cdot 10^5$) — the parameters of the j-th query.

Output

For each query, print one integer — the maximum value of $c(v) - m \cdot k$ you can achieve.

Example

input

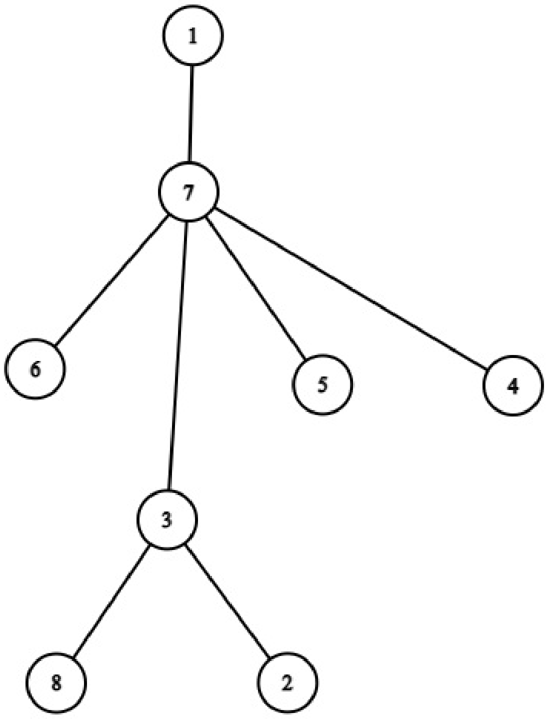
```
8
6 7
3 2
8 3
5 7
7 4
7 1
7 3
6
1 0
1 2
1 3
7 1
5 0
7 200000
```

output

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

Note

The tree in the first example is shown in the following picture:



Answers to the queries are obtained as follows:

1. $v = 1, k = 0$: you can delete vertices 7 and 3, so the vertex 1 has 5 children (vertices 2, 4, 5, 6, and 8), and the score is $5 - 2 \cdot 0 = 5$;
2. $v = 1, k = 2$: you can delete the vertex 7, so the vertex 1 has 4 children (vertices 3, 4, 5, and 6), and the score is $4 - 1 \cdot 2 = 2$.
3. $v = 1, k = 3$: you shouldn't delete any vertices, so the vertex 1 has only one child (vertex 7), and the score is $1 - 0 \cdot 3 = 1$;
4. $v = 7, k = 1$: you can delete the vertex 3, so the vertex 7 has 5 children (vertices 2, 4, 5, 6, and 8), and the score is $5 - 1 \cdot 1 = 4$;
5. $v = 5, k = 0$: no matter what you do, the vertex 5 will have no children, so the score is 0;
6. $v = 7, k = 200000$: you shouldn't delete any vertices, so the vertex 7 has 4 children (vertices 3, 4, 5, and 6), and the score is $4 - 0 \cdot 200000 = 4$.