Codeforces Round #629 (Div. 3)

A. Divisibility Problem

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

You are given two positive integers a and b. In one move you can increase a by (replace a with a). Your task is to find the minimum number of moves you need to do in order to make a divisible by b. It is possible, that you have to make moves, as a is already divisible by b. You have to answer t independent test cases.

Input

The first line of the input contains one integer t (t) — the number of test cases. Then t test cases follow.

The only line of the test case contains two integers a and b (a b).

Output

For each test case print the answer — the minimum number of moves you need to do in order to make a divisible by b.

Example

input

```
5
10 4
13 9
100 13
123 456
92 46
```

output

```
2
5
4
333
0
```

B. K-th Beautiful String

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

For the given integer n (n) let's write down all the strings of length n which contain n letters 'a' and two letters 'b' in **lexicographical** (alphabetical) order.

Recall that the string s of length n is lexicographically less than string t of length n, if there exists such i (i n), that s_i t_i, and for any j (j i) s_j t_j. The lexicographic comparison of strings is implemented by the operator < in modern programming languages.

For example, if n the strings are (the order does matter):

1. aaabb
2. aabab
3. aabba
4. abaab
5. ababa
6. abbaa
7. baaab
8. baaba
9. babaa
10. bbaaa

It is easy to show that such a list of strings will contain exactly $\frac{n \quad n}{\quad}$ strings.

You are given n (n) and k (k $\frac{n \quad n}{\quad}$). Print the k-th string from the list.

Input

The input contains one or more test cases.

The first line contains one integer t (t) — the number of test cases in the test. Then t test cases follow.

Each test case is written on the the separate line containing two integers n and k (n k $\frac{n\ n}{}$.

The sum of values n over all test cases in the test doesn't exceed .

Output

For each test case print the k-th string from the list of all described above strings of length n. Strings in the list are sorted lexicographically (alphabetically).

Example

input

```
7
5 1
5 2
5 8
5 10
3 1
3 2
20 100
```

output

```
aaabb
aabab
baaba
bbaaa
abb
bab
aaaaabaaaaabaaaaaaaa
```

C. Ternary XOR

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

A number is *ternary* if it contains only digits , and . For example, the following numbers are ternary: , , , .

You are given a long ternary number x. The first (leftmost) digit of x is guaranteed to be , the other digits of x can be , or .

Let's define the ternary XOR operation of two ternary numbers a and b (both of length n) as a number c a b of length n, where c_i a_i b_i (where is modulo operation). In other words, add the corresponding digits and take the remainders of the sums when divided by . For example, .

Your task is to find such ternary numbers a and b both of length n and both without leading zeros that a b x and max a b is the minimum possible.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t (t) — the number of test cases. Then t test cases follow. The first line of the test case contains one integer n (n) — the length of x. The second line of the test case contains ternary number x consisting of n digits or . It is guaranteed that the first digit of x is . It is guaranteed that the sum of n over all test cases does not exceed (n).

Output

For each test case, print the answer — two ternary integers a and b both of length n and both without leading zeros such that a b x and max a b is the minimum possible. If there are several answers, you can print any.

Example

input

```
4
5
22222
5
21211
1
2
9
220222021
```

output

```
11111
11111
11000
10211
1
```

```
1
110111011
110111010
```

D. Carousel

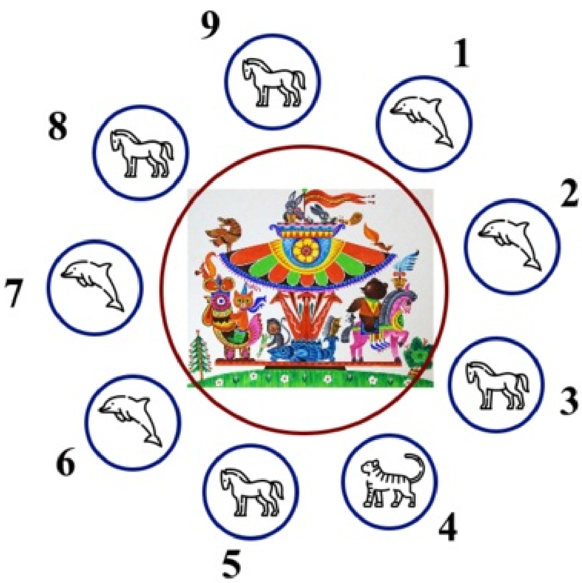
time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The round carousel consists of n figures of animals. Figures are numbered from to n in order of the carousel moving. Thus, after the n-th figure the figure with the number follows. Each figure has its own type — the type of the animal corresponding to this figure (the horse, the tiger and so on). The type of animal of the i-th figure equals t_i.



The example of the carousel for n and t .

You want to color each figure in one of the colors. You think that it's boring if the carousel contains two different figures (with the distinct types of animals) going one right after another and colored in the same color.

Your task is to color the figures in such a way that the number of distinct colors used is the minimum possible and there are no figures of the different types going one right after another and colored in the same color. If you use exactly k distinct colors, then the colors of figures should be denoted with integers from to k.

Input

The input contains one or more test cases.

The first line contains one integer q (q) — the number of test cases in the test. Then q test cases follow. One test case is given on two lines.

The first line of the test case contains one integer n (n) — the number of figures in the carousel. Figures are numbered from to n in order of carousel moving. Assume that after the n-th figure the figure goes.

The second line of the test case contains n integers t t t_n (t_i), where t_i is the type of the animal of the i-th figure.

The sum of n over all test cases does not exceed .

Output

Print q answers, for each test case print two lines.

In the first line print one integer k — the minimum possible number of distinct colors of figures.

In the second line print n integers c c c_n (c_i k), where c_i is the color of the i-th figure. If there are several answers, you can print any.

Example

input

```
4
5
1 2 1 2 2
6
1 2 2 1 2 2
5
1 2 1 2 3
3
10 10 10
```

output

```
2
1 2 1 2 2
2
2 1 2 1 2 1
3
2 3 2 3 1
1
1 1 1
```

E. Tree Queries

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a rooted tree consisting of n vertices numbered from to n. The root of the tree is a vertex number .

A tree is a connected undirected graph with n edges.

You are given m queries. The i-th query consists of the set of k_i distinct vertices v_i v_i v_i k_i . Your task is to say if there is a path from the root to some vertex u such that each of the given k vertices is either belongs to this path or has the distance to some vertex of this path.

Input

The first line of the input contains two integers n and m (n , m) — the number of vertices in the tree and the number of queries.

Each of the next n lines describes an edge of the tree. Edge i is denoted by two integers u_i and v_i, the labels of vertices it connects u_i v_i n u_i v_i).

It is guaranteed that the given edges form a tree.

The next m lines describe queries. The i-th line describes the i-th query and starts with the integer k_i (k_i n) — the number of vertices in the current query. Then k_i integers follow: v_i v_i v_i k_i (v_i j n), where v_i j is the j-th vertex of the i-th query.

It is guaranteed that all vertices in a single query are distinct.

It is guaranteed that the sum of k_i does not exceed ($\sum_{i}^{m} k_i$).

Output

For each query, print the answer — "YES", if there is a path from the root to some vertex u such that each of the given k vertices is either belongs to this path or has the distance to some vertex of this path and "NO" otherwise.

Example

input

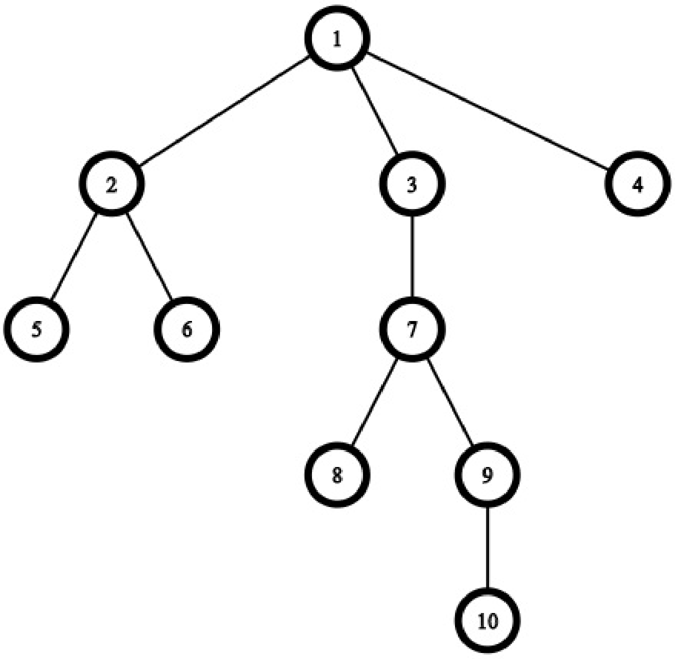
```
10 6
1 2
1 3
1 4
2 5
2 6
3 7
7 8
7 9
9 10
4 3 8 9 10
3 2 4 6
3 2 1 5
3 4 8 2
2 6 10
3 5 4 7
```

output

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

Note

The picture corresponding to the example:



Consider the queries.

The first query is . The answer is "YES" as you can choose the path from the root to the vertex u . Then vertices belong to the path from to and the vertex has distance to the vertex which also belongs to this path.

The second query is . The answer is "YES" as you can choose the path to the vertex u . Then the vertex has distance to the vertex which belongs to this path and the vertex has distance to the vertex which belongs to this path.

The third query is . The answer is "YES" as you can choose the path to the vertex u and all vertices of the query belong to this path.

The fourth query is . The answer is "YES" as you can choose the path to the vertex u so vertices and both have distance to the vertex which belongs to this path and the vertex has distance to the vertex which belongs to this path.

The fifth and the sixth queries both have answer "NO" because you cannot choose suitable vertex u.

F. Make k Equal

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

You are given the array a consisting of n elements and the integer k n.

You want to obtain **at least** k equal elements in the array a. In one move, you can make one of the following two operations:

- Take **one** of the minimum elements of the array and increase its value by one (more formally, if the minimum value of a is mn then you choose such index i that a_i mn and set a_i a_i);
- take **one** of the maximum elements of the array and decrease its value by one (more formally, if the maximum value of a is mx then you choose such index i that a_i mx and set a_i a_i).

Your task is to calculate the minimum number of moves required to obtain **at least** k equal elements in the array.

Input

The first line of the input contains two integers n and k (k n) — the number of elements in a and the required number of equal elements.

The second line of the input contains n integers a a a_n (a_i), where a_i is the i-th element of a.

Output

Print one integer — the minimum number of moves required to obtain **at least** k equal elements in the array.

Examples

input
6 5 1 2 2 4 2 3

output
3

input

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

output

```
4
```