A. Consecutive Sum Riddle

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

Theofanis has a riddle for you and if you manage to solve it, he will give you a Cypriot snack halloumi for free (Cypriot cheese).

You are given an integer n. You need to find two integers l and r such that $-10^{18} \le l < r \le 10^{18}$ and $l + (l+1) + \ldots + (r-1) + r = n$.

Input

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

The first and only line of each test case contains a single integer n ($1 \le n \le 10^{18}$).

Output

For each test case, print the two integers l and r such that $-10^{18} \le l < r \le 10^{18}$ and $l + (l+1) + \ldots + (r-1) + r = n$.

It can be proven that an answer always exists. If there are multiple answers, print any.

Example

input

```
7
1
2
3
6
100
25
3000000000000
```

output

```
0 1
-1 2
1 2
1 3
18 22
-2 7
999999999999 1000000000001
```

Note

In the first test case, $0 + 1 = 1$.

In the second test case, $(-1) + 0 + 1 + 2 = 2$.

In the fourth test case, $1 + 2 + 3 = 6$.

In the fifth test case, $18 + 19 + 20 + 21 + 22 = 100$.

In the sixth test case, $(-2) + (-1) + 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 = 25$.

B. Special Numbers

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

Theofanis really likes sequences of positive integers, thus his teacher (Yeltsa Kcir) gave him a problem about a sequence that consists of only special numbers.

Let's call a positive number *special* if it can be written as a sum of **different** non-negative powers of n. For example, for $n = 4$ number 17 is special, because it can be written as $4^0 + 4^2 = 1 + 16 = 17$, but 9 is not.

Theofanis asks you to help him find the k-th special number if they are sorted in increasing order. Since this number may be too large, output it modulo $10^9 + 7$.

Input

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

The first and only line of each test case contains two integers n and k ($2 \le n \le 10^9$; $1 \le k \le 10^9$).

Output

For each test case, print one integer — the k-th special number in increasing order modulo $10^9 + 7$.

Example

input

```
3
3 4
2 12
105 564
```

output

```
9
12
3595374
```

Note

For $n = 3$ the sequence is $[1, 3, 4, 9...]$

C. Make Them Equal

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

Theofanis has a string $s_1 s_2 \dots s_n$ and a character c. He wants to make all characters of the string equal to c using the minimum number of operations.

In one operation he can choose a number x ($1 \le x \le n$) and **for every position** i, where i is **not** divisible by x, replace s_i with c.

Find the minimum number of operations required to make all the characters equal to c and the x-s that he should use in his operations.

Input

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

The first line of each test case contains the integer n ($3 \le n \le 3 \cdot 10^5$) and a lowercase Latin letter c — the length of the string s and the character the resulting string should consist of.

The second line of each test case contains a string s of lowercase Latin letters — the initial string.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, firstly print one integer m — the minimum number of operations required to make all the characters equal to c.

Next, print m integers $x_1, x_2, \dots, x_m$ ($1 \le x_j \le n$) — the x-s that should be used in the order they are given.

It can be proved that under given constraints, an answer always exists. If there are multiple answers, print any.

Example

input

```
3
4 a
aaaa
4 a
baaa
4 b
bzyx
```

output

```
0
1
2
2
2 3
```

Note

Let's describe what happens in the third test case:

1. $x_1 = 2$: we choose all positions that are not divisible by 2 and replace them, i. e. bzyx $\rightarrow$ bzbx;
2. $x_2 = 3$: we choose all positions that are not divisible by 3 and replace them, i. e. bzbx $\rightarrow$ bbbb.

D. The Number of Imposters

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

Theofanis started playing the new online game called "Among them". However, he always plays with Cypriot players, and they all have the same name: "Andreas" (the most common name in Cyprus).

In each game, Theofanis plays with n other players. Since they all have the same name, they are numbered from 1 to n.

The players write m comments in the chat. A comment has the structure of "i j c" where i and j are two distinct integers and c is a string ($1 \le i, j \le n$; $i \ne j$; c is either `imposter` or `crewmate`). The comment means that player i said that player j has the role c.

An imposter always lies, and a crewmate always tells the truth.

Help Theofanis find the maximum possible number of imposters among all the other Cypriot players, or determine that the comments contradict each other (see the notes for further explanation).

Note that each player has exactly **one** role: either `imposter` or `crewmate`.

Input

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

The first line of each test case contains two integers n and m ($1 \le n \le 2 \cdot 10^5$; $0 \le m \le 5 \cdot 10^5$) — the number of players except Theofanis and the number of comments.

Each of the next m lines contains a comment made by the players of the structure "i j c" where i and j are two **distinct** integers and c is a string ($1 \le i, j \le n$; $i \ne j$; c is either `imposter` or `crewmate`).

There can be multiple comments for the same pair of (i, j).

It is guaranteed that the sum of all n does not exceed $2 \cdot 10^5$ and the sum of all m does not exceed $5 \cdot 10^5$.

Output

For each test case, print one integer — the **maximum** possible number of imposters. If the comments contradict each other, print -1.

Example

input

```
5
3 2
1 2 imposter
2 3 crewmate
5 4
1 3 crewmate
2 5 crewmate
2 4 imposter
3 4 imposter
2 2
1 2 imposter
2 1 crewmate
3 5
1 2 imposter
1 2 imposter
3 2 crewmate
3 2 crewmate
1 3 imposter
5 0
```

output

```
2
4
-1
2
5
```

Note

In the first test case, imposters can be Andreas 2 and 3.

In the second test case, imposters can be Andreas 1, 2, 3 and 5.

In the third test case, comments contradict each other. This is because player 1 says that player 2 is an imposter, and player 2 says that player 1 is a crewmate. If player 1 is a crewmate, then he must be telling the truth, so player 2 must be an imposter. But if player 2 is an imposter then he must be lying, so player 1 can't be a crewmate. Contradiction.

E1. Rubik's Cube Coloring (easy version)

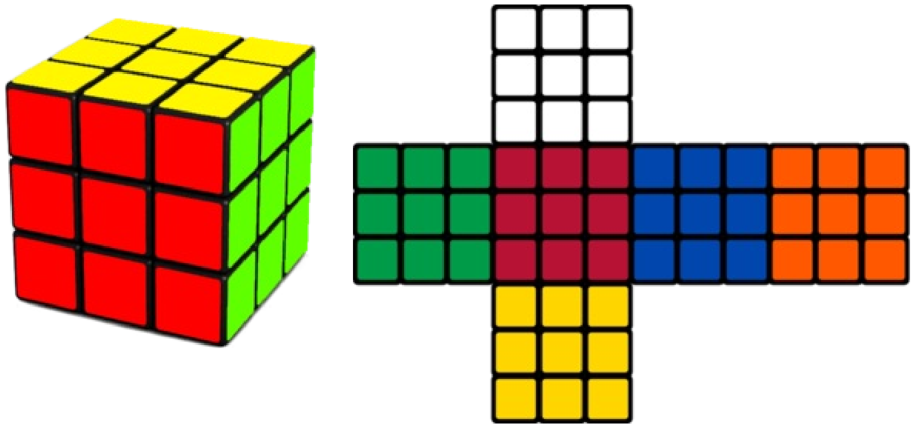
time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

It is the easy version of the problem. The difference is that in this version, there are no nodes with already chosen colors.

Theofanis is starving, and he wants to eat his favorite food, sheftalia. However, he should first finish his homework. Can you help him with this problem?

You have a *perfect* binary tree of $2^k - 1$ nodes — a binary tree where all vertices i from 1 to $2^{k-1} - 1$ have exactly two children: vertices $2i$ and $2i + 1$. Vertices from 2^{k-1} to $2^k - 1$ don't have any children. You want to color its vertices with the 6 Rubik's cube colors (White, Green, Red, Blue, Orange and Yellow).

Let's call a coloring *good* when all edges connect nodes with colors that are *neighboring* sides in the Rubik's cube.



A picture of Rubik's cube and its 2D map.

More formally:

- a white node can **not** be neighboring with white and yellow nodes;
- a yellow node can **not** be neighboring with white and yellow nodes;
- a green node can **not** be neighboring with green and blue nodes;
- a blue node can **not** be neighboring with green and blue nodes;
- a red node can **not** be neighboring with red and orange nodes;
- an orange node can **not** be neighboring with red and orange nodes;

You want to calculate the number of the good colorings of the binary tree. Two colorings are considered different if at least one node is colored with a different color.

The answer may be too large, so output the answer modulo $10^9 + 7$.

Input

The first and only line contains the integers k ($1 \leq k \leq 60$) — the number of levels in the perfect binary tree you need to color.

Output

Print one integer — the number of the different colorings modulo $10^9 + 7$.

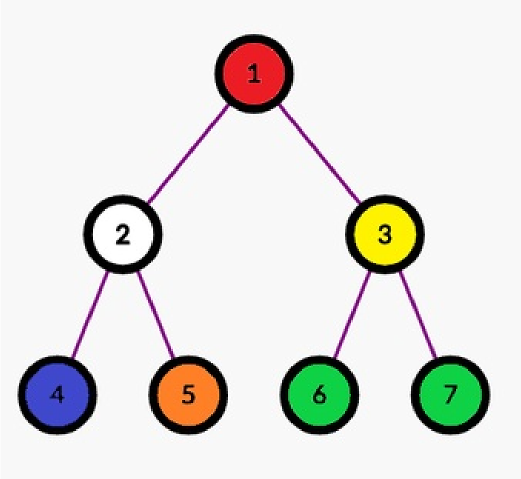
Examples

input
3
output
24576

input
14
output
934234

Note

In the picture below, you can see one of the correct colorings of the first example.



E2. Rubik's Cube Coloring (hard version)

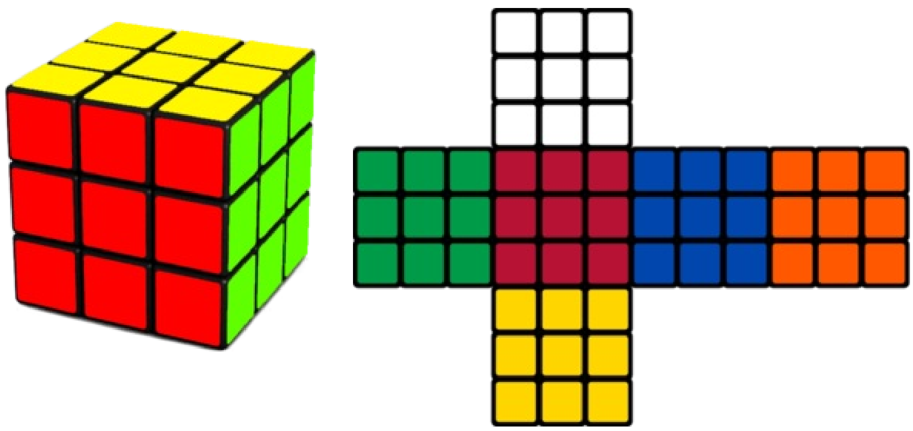
time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

It is the hard version of the problem. The difference is that in this version, there are nodes with already chosen colors.

Theofanis is starving, and he wants to eat his favorite food, sheftalia. However, he should first finish his homework. Can you help him with this problem?

You have a *perfect* binary tree of $2^k - 1$ nodes — a binary tree where all vertices i from 1 to $2^{k-1} - 1$ have exactly two children: vertices $2i$ and $2i + 1$. Vertices from 2^{k-1} to $2^k - 1$ don't have any children. You want to color its vertices with the 6 Rubik's cube colors (White, Green, Red, Blue, Orange and Yellow).

Let's call a coloring *good* when all edges connect nodes with colors that are *neighboring* sides in the Rubik's cube.



A picture of Rubik's cube and its 2D map.

More formally:

- a white node can **not** be neighboring with white and yellow nodes;
- a yellow node can **not** be neighboring with white and yellow nodes;
- a green node can **not** be neighboring with green and blue nodes;
- a blue node can **not** be neighboring with green and blue nodes;
- a red node can **not** be neighboring with red and orange nodes;
- an orange node can **not** be neighboring with red and orange nodes;

However, there are n special nodes in the tree, colors of which are already chosen.

You want to calculate the number of the good colorings of the binary tree. Two colorings are considered different if at least one node is colored with a different color.

The answer may be too large, so output the answer modulo $10^9 + 7$.

Input

The first line contains the integers k $(1 \le k \le 60)$ — the number of levels in the perfect binary tree you need to color.

The second line contains the integer n $(1 \le n \le \min(2^k - 1, 2000))$ — the number of nodes, colors of which are already chosen.

The next n lines contains integer v $(1 \le v \le 2^k - 1)$ and string s — the index of the node and the color of the node (s is one of the `white`, `yellow`, `green`, `blue`, `red` and `orange`).

It is guaranteed that each node v appears in the input at most once.

Output

Print one integer — the number of the different colorings modulo $10^9 + 7$.

Examples

input

```
3
2
5 orange
2 white
```

output

```
1024
```

input

```
2
2
1 white
2 white
```

output

```
0
```

input

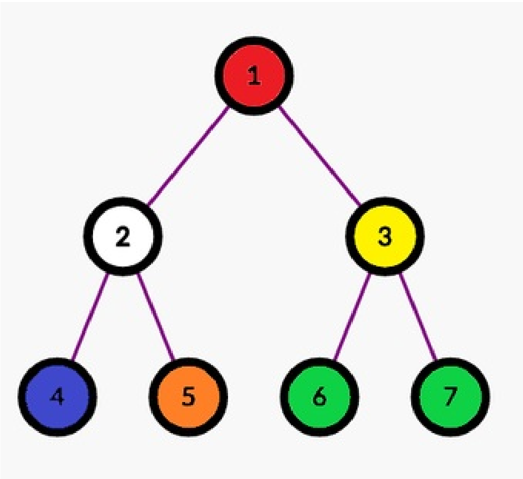
```
10
3
1 blue
4 red
5 orange
```

output

```
328925088
```

Note

In the picture below, you can see one of the correct colorings of the first test example.



F. Ideal Farm

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

Theofanis decided to visit his uncle's farm. There are s animals and n animal pens on the farm. For utility purpose, animal pens are constructed in one row.

Uncle told Theofanis that a farm is *lucky* if you can distribute all animals in all pens in such a way that there are no empty pens and there is at least one continuous segment of pens that has exactly k animals in total.

Moreover, a farm is *ideal* if it's lucky for any distribution without empty pens.

Neither Theofanis nor his uncle knows if their farm is ideal or not. Can you help them to figure it out?

Input

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

The first and only line of each test case contains three integers s, n, and k ($1 \le s, n, k \le 10^{18}$; $n \le s$).

Output

For each test case, print YES (case-insensitive), if the farm is ideal, or NO (case-insensitive) otherwise.

Example

input

```
4
1 1 1
1 1 2
100 50 200
56220 47258 14497
```

output

```
YES
NO
NO
YES
```

Note

For the first and the second test case, the only possible combination is $[1]$ so there always will be a subsegment with 1 animal but not with 2 animals.