Meeting in the Middle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

The Boiler's Land is a nation of N cities. Some cities are connected via bidirectional roads. The i-th road has a weight limit of w_i pounds, which means that any vehicle travelling on it cannot exceed w_i pounds. The i-th road also has a toll of c_i dollars, which means that any vehicle passing through it must pay c_i dollars.

Alice, Bob and Charlie live in cities A, B and C respectively. They have decided to hold a music festival in one of the N cities during the winter break. They will each carry some of the equipment necessary for the festival to the site.

Unfortunately for them, the equipment is so heavy that some of the roads might not be available for some of them due to the weight limit. They made some measurements and found out that their cars weighed w_A, w_B and w_C respectively after being loaded with the equipment they planned to carry.

Given the limitation, they think that they are only capable of holding the festival in some of the cities. Furthermore, they want to minimize the toll they pay as it counts towards the budget they have. Can you help them figure out the best city to hold the festival?

Input

The first line contains N and M, the number of the cities and the number of the roads, respectively.

The next line contains A, B, C, w_A, w_B and w_C, respectively, as described in the problem statement.

The next M lines each contain the description of an edge u_i, v_i, w_i and c_i, respectively, meaning there is an edge between u_i and v_i with a weight limit of w_i pounds and a toll of c_i dollars.

There may be **multiple edges between two cities.**

Output

Two integers S and C, meaning that the festival should be held in city S and the total toll would be C dollars. If there are multiple cities that minimize the total toll, output the city with the smallest index. If there are no cities that satisfy the constraint, output -1.

Examples

standard input	standard output

`4 3`
`1 2 3 10000 20000 30000`
`1 4 10000 10`
`2 4 20000 10`
`3 4 30000 10`
	`4 30`

`4 3`
`1 2 3 10000 20000 30000`
`1 4 10000 10`
`2 4 10000 10`
`3 4 10000 10`
	`-1`

`2 1`
`1 1 1 10000 20000 30000`
`1 2 10000 10`
	`1 0`

Note

$1 \le N \le 10^5, 1 \le M \le 5 \times 10^5$.

$1 \le A, B, C \le N$.

$1 \le w_A, w_B, w_C \le 10^9$.

$1 \le u_i, v_i \le N, u_i \ne v_i$.

$1 \le w_i \le 10^9, 1 \le c_i \le 10^9$.

USACO 2023 US OPEN CONTEST, SILVER
PROBLEM 2. FIELD DAY

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

****Note: The time limit for this problem in Python is 15s. Other languages have the default time limit of 2s.****

Each of Farmer John's N barns $(2 \le N \le 10^5)$ has selected a team of C cows $(1 \le C \le 18)$ to participate in field day. The breed of every cow is either a Guernsey or a Holstein.

The difference between two teams is defined to be the number of positions i $(1 \le i \le C)$ at which the breeds of the cows in the ith positions differ. For every team t from $1 \dots N$, please compute the maximum difference between team t and any other team.

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains C and N.

The next N lines each contain a string of length C of Gs and Hs. Each line corresponds to a team.

OUTPUT FORMAT (print output to the terminal / stdout):

For each team, print the maximum difference.

SAMPLE INPUT:

```
5 3
GHGGH
GHHHH
HGHHG
```

SAMPLE OUTPUT:

```
5
3
5
```

The first and third teams differ by 5. The second and third teams differ by 3.

SCORING:

- Inputs 2-5: $C = 10$
- Inputs 6-9: All answers are at least $C - 3$.
- Inputs 10-20: No additional constraints.

Problem credits: Benjamin Qi

Contest has ended. No further submissions allowed.

Snapper Chain (Hard)

Note that this is a harder version of the problem snappereasy

The *Snapper* is a clever little device that, on one side, plugs its input plug into an output socket, and, on the other side, exposes an output socket for plugging in a light or other device.

When a *Snapper* is in the ON state and is receiving power from its input plug, then the device connected to its output socket is receiving power as well. When you snap your fingers – making a clicking sound – any *Snapper* receiving power at the time of the snap toggles between the ON and OFF states.

In hopes of destroying the universe by means of a singularity, I have purchased N *Snapper* devices and chained them together by plugging the first one into a power socket, the second one into the first one, and so on. The light is plugged into the Nth *Snapper*.

Initially, all the *Snapper*s are in the OFF state, so only the first one is receiving power from the socket, and the light is off. I snap my fingers once, which toggles the first *Snapper* into the ON state and gives power to the second one. I snap my fingers again, which toggles both *Snapper*s and then promptly cuts power off from the second one, leaving it in the ON state, but with no power. I snap my fingers the third time, which toggles the first *Snapper* again and gives power to the second one. Now both *Snapper*s are in the ON state, and if my light is plugged into the second *Snapper* it will be *on*.

I keep doing this for hours. Will the light be *on* or *off* after I have snapped my fingers K times? The light is *on* if and only if it's receiving power from the *Snapper* it's plugged into.

Input

The first line of the input gives the number of test cases, T. T lines follow. Each one contains two integers, N and K.

You may assume that $1 \leq T \leq 10\,000$, $1 \leq N \leq 30$ and $0 \leq K \leq 10^8$.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is either "ON" or "OFF", indicating the state of the light bulb.

Sample Input 1

```
4
1 0
1 1
4 0
4 47
```

Sample Output 1

```
Case #1: OFF
Case #2: ON
Case #3: OFF
Case #4: ON
```

Metadata

Hide

CPU Time limit	1 second
Memory limit	1024 MB
Difficulty	2.5 Easy
Source	Google Code Jam 2010 Qualification Round
License	(cc) BY