

# Google Code Jam '14 Round 1A Problem A - Charging Chaos

**Time limit:** 120.0s    **Memory limit:** 1G

Shota the farmer has a problem. He has just moved into his newly built farmhouse, but it turns out that the outlets haven't been configured correctly for all of his devices. Being a modern farmer, Shota owns a large number of smartphones and laptops, and even owns a tablet for his favorite cow Wagyu to use. In total, he owns  $N$  different devices.

As these devices have different specifications and are made by a variety of companies, they each require a different electric flow to charge. Similarly, each outlet in the house outputs a specific electric flow. An electric flow can be represented by a string of 0s and 1s of length  $L$ .

Shota would like to be able to charge all  $N$  of his devices at the same time. Coincidentally, there are exactly  $N$  outlets in his new house. In order to configure the electric flow from the outlets, there is a master control panel with  $L$  switches. The  $i^{\text{th}}$  switch flips the  $i^{\text{th}}$  bit of the electric flow from each outlet in the house. For example, if the electric flow from the outlets is:

```
Outlet 0: 10
Outlet 1: 01
Outlet 2: 11
```

Then flipping the second switch will reconfigure the electric flow to:

```
Outlet 0: 11
Outlet 1: 00
Outlet 2: 10
```

If Shota has a smartphone that needs flow `11` to charge, a tablet that needs flow `10` to charge, and a laptop that needs flow `00` to charge, then flipping the second switch will make him very happy!

Misaki has been hired by Shota to help him solve this problem. She has measured the electric flows from the outlets in the house, and noticed that they are all different. Decide if it is possible for Shota to charge all of his devices at the same time, and if it is possible, figure out the minimum number of switches that needs to be flipped, because the switches are big and heavy and Misaki doesn't want to flip more than what she needs to.

## Input Specification

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow. Each test case consists of three lines. The first line contains two space-separated integers  $N$  and  $L$ . The second line contains  $N$  space-separated strings of

length  $L$ , representing the initial electric flow from the outlets. The third line also contains  $N$  space-separated strings of length  $L$ , representing the electric flow required by Shota's devices.

## Output Specification

---

For each test case, output one line containing `Case #x: y`, where  $x$  is the case number (starting from 1) and  $y$  is the minimum number of switches to be flipped in order for Shota to charge all his devices. If it is impossible,  $y$  should be the string `NOT POSSIBLE` (without the quotes). Please note that our judge is not case-sensitive, but it is strict in other ways: so although `not possible` will be judged correct, any misspelling will be judged wrong. We suggest copying/pasting the string `NOT POSSIBLE` into your code.

## Limits

---

Memory limit: 1 GB.

$$1 \leq T \leq 100.$$

No two outlets will be producing the same electric flow, initially.

No two devices will require the same electric flow.

### Small Dataset

Time limit: 60 seconds.

$$1 \leq N \leq 10.$$

$$2 \leq L \leq 10.$$

### Large Dataset

Time limit: 120 seconds.

$$1 \leq N \leq 150.$$

$$10 \leq L \leq 40.$$

## Sample Input

---

```
3
3 2
01 11 10
11 00 10
2 3
101 111
010 001
2 2
01 10
10 01
```

## Sample Output

---

```
Case #1: 1
Case #2: NOT POSSIBLE
Case #3: 0
```

## Explanation for Sample

---

In the first example case, Misaki can flip the second switch once. The electric flow from the outlets becomes:

```
Outlet 0: 00
Outlet 1: 10
Outlet 2: 11
```

Then Shota can use the outlet 0 to charge device 1, the outlet 1 to charge device 2, outlet 2 to charge device 0. This is also a solution that requires the minimum amount number of switches to be flipped.

## Note

---

This problem has different time limits for different batches. If you exceed the Time Limit for any batch, the judge will incorrectly display `>120.000s` regardless of the actual time taken. Refer to the **Limits** section for batch-specific time limits.