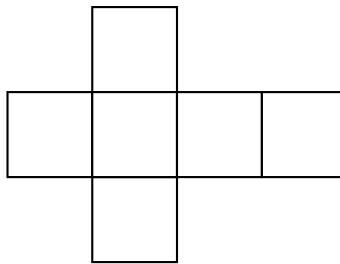


Alice is very fond of building up paper cubes using their nets cut out of paper. She draws the outlines of nets on papers, cuts them out along the outlines, and folds them into cubes. Given below is an example for a cube net:



While drawing them, she finds out that there are many different ways of drawing the net of a cube, and that only some of the "nets" she drew could actually be folded into a cube.

Can you help Alice to determine which of the outlines drawn on the paper can be folded into cubes, before cutting them out?

**Task:** In each testcase, you are given a square-ruled paper of a given size  $m \times n$  (matrix notation), where each character represents a single square. Their notation is as follows:

- A square belonging to a net candidate is denoted by a '#'.
- A square not belonging to a net candidate is denoted with a '.'.

Print the number of valid nets and invalid nets separated by a space.

## Input Format

The first line contains a positive integer  $q$ , that denotes the number of queries.

The subsequent lines describe each query in the following format:

- The first line contains two space-separated integers  $m, n$  that denote the size of the paper (matrix notation).
- Each of the next  $m$  lines contains  $n$  space-separated characters that describe the arrangement of the nets on the paper: '#' for a square in a net candidate, and '.' for a square not in a net candidate.

## Constraints

$$1 \leq q \leq 100$$

$$1 \leq m, n \leq 1000$$

Other constraints:

- There will not be any net candidates with **more than** 6 squares.

## Output Format

The output should have  $q$  lines each consisting of two space-separated integers  $p$  and  $q$ , with  $p$  denoting the number of valid nets, and  $q$  denoting the number of invalid nets of the corresponding query.

Sample Input 0

```
1
6 5
.....
..#..
.###.
..#..
..#..
.....
```

Sample Output 0

```
1 0
```

Explanation 0

There is a single query in this testcase, which has a grid of size  $6 \times 5$ , in which a single net candidate is present. Here, the given net candidate is indeed a valid net as it can be folded into a cube. There are no invalid nets.

Therefore, the numbers of valid nets and invalid nets are 1 and 0 respectively.

Sample Input 1

```
3
10 10
.....
.....
.....
...#....
...###...
...#....
...#....
...#....
.....
.....
.....
7 6
.....
...#..
...#..
...##..
...#...
...#...
.....
8 3
...
.#.
.#.
.#.
.#.
.#.
.#.
...

```

Sample Output 1

```
1 0
1 0
0 1
```

### Explanation 1

Here there are three queries, each having a single net candidate.

- **Queries 1, 2:** The given net can be folded into a cube; hence it is a valid net. There are no invalid nets. Hence, the output is '1 0' for both these queries.
- **Query 3:** The given net candidate cannot be folded into a cube; hence it is an invalid net. There are no valid nets. Hence, the output is '0 1'.