

## 4034 Game of Life

LIFE is a evolutionary game played on a 2D game board. Initially, the game board is filled with white and black stones. For each iteration of the game, each stone is checked and perhaps changed to the other color according to the rules given below:

1. For each stone  $i$ , if there are more black stones than white stones in the  $3 \times 3$  neighborhood centered at stone  $i$ , then turn stone  $i$  into a black stone, otherwise, turn it to a white stone. The checking is based on the **previous** iterated game board configuration, and **NOT** the current iteration of intermediate game board. In other words, the check of all stones are done simultaneously. Thus, changing the stone color will not affect any other stone in the same iteration.
2. The boundary of the game board will remain unchanged throughout all iterations. In other words, there is no need to check the stones located on the boundary of the game board.

Please write a program that when given an initial game board configuration and the number of iterations of LIFE, compute and output the number of black and white stones on the resulting game board.

### Technical Specification

1. The game board size is  $m \times m$  where  $3 \leq m \leq 512$ .
2. The number of iterations is  $t$ , where  $1 \leq t \leq 100$ .

### Input

The first line of the input contains an integer  $n$  indicating the number of test cases to follow.

For each test case, the first line contains two integers,  $m$  and  $t$ , specifying the game board dimension and number of LIFE iterations. The next  $m$  lines outline the configuration of the initial  $m \times m$  game board. Each line contains  $m$  consecutive characters, where each character is either 'b' or 'w' denoting black stone or white stone, respectively.

### Output

For each test case, output on a single line the number of black stones and white stones of the resulting game board.

**Note:** The final game board configuration for the two cases below are:

```
wbwb  
bbbw  
wbbb  
wbwb
```

```
wbbbb  
wwbww  
wwwww  
wwwww  
wwwww
```

**Sample Input**

```
2
4 1
wbwb
bbbw
wwbb
wbwb
5 1
wbwww
wbwbw
wwwww
wwwww
wwwww
```

**Sample Output**

```
10 6
5 20
```