City Game

Bob is a strategy game programming specialist. In his new city building game the gaming environment is as follows: a city is built up by areas, in which there are streets, trees, factories and buildings. There is still some space in the area that is unoccupied. The strategic task of his game is to win as much rent money from these free spaces. To win rent money you must erect buildings, that can only be rectangular, as long and wide as you can. Bob is trying to find a way to build the biggest possible building in each area. But he comes across some problems - he is not allowed to destroy already existing buildings, trees, factories and streets in the area he is building in.

Each area has its width and length. The area is divided into a grid of equal square units.The rent paid for each unit on which you're building stands is 3$.

Your task is to help Bob solve this problem. The whole city is divided into **K** areas. Each one of the areas is rectangular and has a different grid size with its own length **M** and width **N**. The existing occupied units are marked with the symbol **R**. The unoccupied units are marked with the symbol **F**.

Input

The first line of the input contains an integer **K** - determining the number of datasets. Next lines contain the area descriptions. One description is defined in the following way: The first line contains two integers-area length **M**<=1000 and width **N**<=1000, separated by a blank space. The next **M** lines contain **N** symbols that mark the reserved or free grid units,separated by a blank space. The symbols used are:

R - reserved unit

F - free unit

In the end of each area description there is a separating line.

Output

For each data set in the input print on a separate line, on the standard output, the integer that represents the profit obtained by erecting the largest building in the area encoded by the data set.

Example

Input:

```
2
5 6
R F F F F F
F F F F F F
R R R F F F
F F F F F F
F F F F F F

5 5
R R R R R
```

R R R R R
R R R R R
R R R R R
R R R R R

Output:
45
0