P25

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 file 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 $M \leq 1000$ and width $N \leq 1000$, 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 file 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.

Sample 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
```

Sample Output

```
45
0
```

P26

In the good old days when Swedish children were still allowed to blow up their fingers with fire-crackers, gangs of excited kids would plague certain smaller cities during Easter time, with only one thing in mind: To blow things up. Small boxes were easy to blow up, and thus mailboxes became a popular target. Now, a small mailbox manufacturer is interested in how many fire-crackers his new mailbox prototype can withstand without exploding and has hired you to help him. He will provide you with k $(1 \leq k \leq 10)$ identical mailbox prototypes each fitting up to m $(1 \leq m \leq 100)$ crackers. However, he is not sure of how many fire-crackers he needs to provide you with in order for you to be able to solve his problem, so he asks you. You think for a while and then say: "Well, if I blow up a mailbox I can't use it again, so if you would provide me with only $k = 1$ mailboxes, I would have to start testing with 1 cracker, then 2 crackers, and so on until it finally exploded. In the worst case, that is if it does not blow up even when filled with m crackers, I would need $1 + 2 + 3 + \ldots + m = m * (m + 1)/2$ crackers. If $m = 100$ that would mean more than 5000 fire-crackers!". "That's too many", he replies. "What if I give you more than $k = 1$ mailboxes? Can you find a strategy that requires less crackers?"

Can you? And what is the minimum number of crackers that you should ask him to provide you with?

You may assume the following:

1. If a mailbox can withstand x fire-crackers, it can also withstand $x - 1$ fire-crackers.
2. Upon an explosion, a mailbox is either totally destroyed (blown up) or unharmed, which means that it can be reused in another test explosion.

Note: If the mailbox can withstand a full load of m fire-crackers, then the manufacturer will of course be satisfied with that answer. But otherwise he is looking for the maximum number of crackers that his mailboxes can withstand.

Input

The input starts with a single integer N $(1 \leq N \leq 10)$ indicating the number of test cases to follow. Each test case is described by a line containing two integers: k and m, separated by a single space.

Output

For each test case print one line with a single integer indicating the minimum number of fire-crackers that is needed, in the worst case, in order to figure out how many crackers the mailbox prototype can withstand.

Sample input

```
4
1 10
1 100
3 73
5 100
```

Sample Output

```
55
5050
382
495
```