7255   Land of Farms

Farmer John and his brothers have found a new land. They are so excited and decide to build new farms on the land. The land is a rectangle and consists of $N \times M$ grids. A farm consists of one or more connected grids. Two grids are adjacent if they share a common border, i.e. their Manhattan distance is exactly 1. In a farm, two grids are considered connected if there exist a series of adjacent grids, which also belong to that farm, between them.

Farmer John wants to build as many farms as possible on the new land. It is required that any two farms should not be adjacent. Otherwise, sheep from different farms would fight on the border. This should be an easy task until several ancient farms are discovered.

Each of the ancient farms also consists of one or more connected grids. Due to the respect to the ancient farmers, Farmer John do not want to divide any ancient farm. If a grid from an ancient farm is selected in a new farm, other grids from the ancient farm should also be selected in the new farm. Note that the ancient farms may be adjacent, because ancient sheep do not fight each other.

The problem is a little complicated now. Can you help Farmer John to find a plan with the maximum number of farms?

Input

The first line of input contains a number T indicating the number of test cases ($T \leq 200$).

Each test case starts with a line containing two integers N and M, indicating the size of the land. Each of the following N lines contains M characters, describing the map of the land ($1 \leq N, M \leq 10$). A grid of an ancient farm is indicated by a single digit (`0-9`). Grids with the same digit belong to the same ancient farm. Other grids are denoted with a single character '.'. It is guaranteed that all test cases are valid.

Output

For each test case, output a single line consisting of 'Case #X: Y'. X is the test case number starting from 1. Y is the maximum number of new farms.

Sample Input

```
3
3 4
..3.
023.
.211
2 3
...
...
4 4
1111
1..1
1991
1111
```

Sample Output

```
Case #1: 4
Case #2: 3
Case #3: 1
```