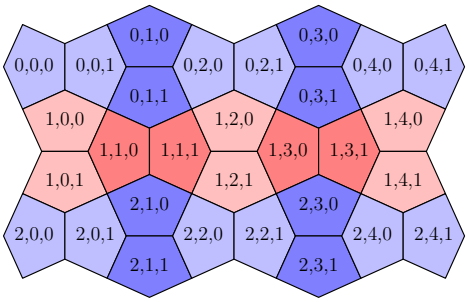
Problem 5 - Bandwidth (`bandwidth`)

The Reply Code Challenge is finally over but our job here is not finished.

After the challenge we have to do all the necessary checks to ensure that everything went smoothly: no submissions were corrupted, no data are lost, and so on.

To execute these difficult tasks we need to run our $\mathbf{ML}^2$ algorithm (Multi-level Machine Learning algorithm) on our complex **5G network** (5-Grid).



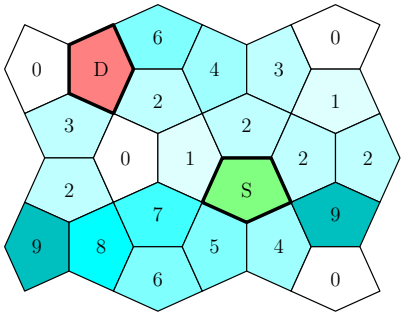
A (3×5) **5G network** representation

A **5G network** is a $n \times m$ grid where each pentagon represents a server. It is possible to index the servers with a triple (i, j, k) where $0 \leq i < n$ is the row, $0 \leq j < m$ is the column and k is 0 for left/up servers or 1 for right/down servers. Following the example of the first figure the network has a total of $2 \times n \times m = 2 \times 3 \times 5 = 30$ servers.

Each server is directly connected with all the servers with which it shares a side, for example $(1, 1, 0)$ is connected with $(0, 1, 1)$, $(1, 0, 0)$, $(1, 0, 1)$, $(1, 1, 1)$ and $(2, 1, 0)$

All the tasks are generated on a specific starting server (in green) and must reach a destination server (in red) using the available connections. Unfortunately, each server has a limited number of tasks, from 0 to 9, that it can manage and exchange in a unit of time, except for the starting and destination servers which can send and receive an unlimited number of tasks.

This lead to a limited bandwidth, i.e. the number of tasks that can be completed in a unit of time, between the starting and the destination server.



As we want to speed up the final checks to announce the winners as soon as possible, Reply approves the budget to upgrade one and only one of the server in the network among those who are not already at the maximum capacity of 9.

Of course, upgrade a single server does not always speed up the the network, so we're asking one last time your support to find a server to upgrade in order to increase the bandwidth.

Input data

The first line of the input file contains an integer **T**, the number of test cases to solve, followed by **T** testcases, numbered from **1** to **T**.

In each test case the first line contains the two integers **N** and **M**, the size of the grid.

The following **N** lines contains a string of length $2 \times M$, each digit of the string represent the capacity of a single server, according to the numerical ordering of the indices. The starting server is marked with a capacity of `S`, the destination with a capacity of `D`.

Output data

The output file must contain **T** lines. For each test case in the input file, the output file must contain a line with the words:

Case #t: i j k

where t is the test case number (from **1** to **T**) and $i\ j\ k$ are the coordinate of the server to upgrade.

If there is no single server to upgrade, i.e. there is no solution, use the following format:

Case #t: -1

Constraints

- $1 \leq \mathbf{T} \leq 16$.
- $1 \leq \mathbf{N}, \mathbf{M} \leq 500$.
- All the capacities are between 0 and 9.

Scoring

- **input 1** : $\mathbf{T} = 1$, $\mathbf{N} = 1$, $\mathbf{M} \leq 5$.
- **input 2** : $\mathbf{T} = 4$, $\mathbf{N} \leq 20$, $\mathbf{M} \leq 20$.
- **input 3** : $\mathbf{T} = 8$, $\mathbf{N} \leq 50$, $\mathbf{M} \leq 50$.
- **input 4** : $\mathbf{T} = 12$, $\mathbf{N} \leq 100$, $\mathbf{M} \leq 100$.
- **input 5** : $\mathbf{T} = 16$, $\mathbf{N} \leq 500$, $\mathbf{M} \leq 500$.

Examples

input	output
2 3 4 0D624301 32012S22 98765490 1 5 S11111111D	Case #1: 1 1 1 Case #2: -1