# Data Structure

## General description

Problems and solutions are stored separately. Problems, which are “blueprints” for games to be played. Every time a player tries to solve a game, a new Attempt is created in which the numbers that the player inserted are saved.

## Problem

The empty map is saved in a double-layered `String` array. Possible values are:

| | |
|---|---|
| # | Not part of the map |
| * | A blocked field, in which no number can be entered |
| ? | An empty field in which the player can enter a number |
| [0-9]* | The value of the field |

Problems can be saved and loaded from a text file. In this case, the first line contains four fields describing the Hidato:

| | | | |
|---|---|---|---|
| Q / T / H | C / CA | [0-9]* | [0-9]* |
| The type of the fields<br>Q = Squares<br>T = Triangles<br>H = Hexagons | The adjacency type<br>C = faces only<br>CA = faces & edges | The number of rows | The number of columns |

The rest of the file is the map that can be played. The contents of the fields are separated by commas. An example:

```
Q,CA,3,4
#,1,?,#
?,?,?,?
7,?,9,#
```

The name of the player who created the problem is saved as `String`. All Problems currently existing in the game are saved in `ProblemCollection`.