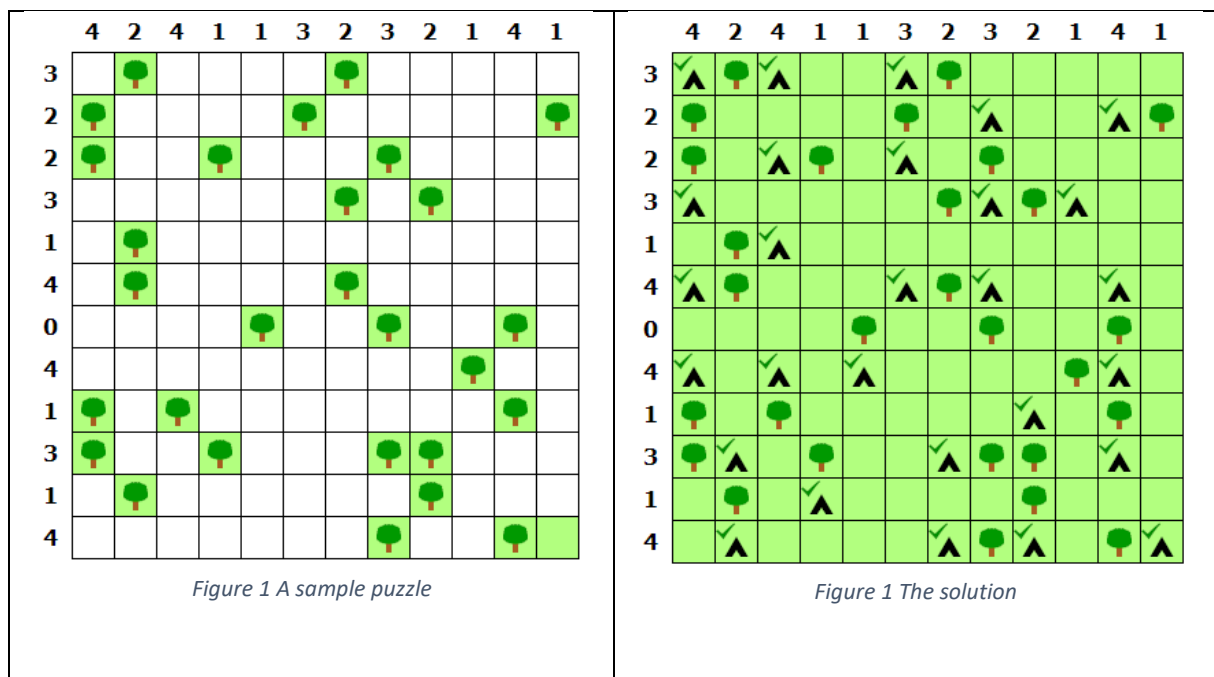


### SE-1108 Self Study-3

You are supposed to implement **TentsPuzzle** class to represent the board of the Tents Puzzle mind game. The objective of the game is to find the places of the tents on a grid where there are a number trees placed. The rules for the placement of the tents are as follows:

- Each tree should have one tent next to it (horizontally or vertically).
- Tents do not touch each other, not even diagonally.
- The numbers outside the grid show how many tents in that row or column

Figure 1 and Figure 2 demonstrates an example puzzle and its solution.



\* The images are taken from : <https://brainbashers.com/tents.asp>

The class should contain 2-dimensional integer array (**board**) and an integer (**puzzleSize**) that stores the size of the puzzle (grid). If the puzzle size is N then the array must be (N+1)X(N+1) (The first row and the first column stores the number of tents on the rows and columns respectively)

In the array except for the first row and the first column; the elements that are 0 represent the empty squares, 1 represents trees 2 represents tents. For example the puzzle demonstrated in Figure-1 is represented by your board array as:

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 2 | 4 | 1 | 1 | 3 | 2 | 3 | 2 | 1 | 4 | 1 |
| 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

In the class implement the following methods:

- Constructor that takes the puzzleSize as parameter and creates an empty board according the puzzleSize ( i.e 2D integer array of size : puzzleSiz+1)
- putTree(int row, int col) : Makes the element at given position tree.
- putTent(int row, int col) : Makes the element at given position tent.
- clear(int row, int col) : Makes the element at given position empty.
- get(int row, int col): returns the element at given position
- toString(): converts the board into string in a user-friendly way:  
For example: Draw vertical horizontal line after the first row and vertical line after the first column to separate number of tents from the actual board. Put "." instead of 0's (empty cells) and "Y" instead of 1's (trees) and \* instead of 2's (tents)
- isValid(checks the board if it is violating any of the rules of Tents puzzle)
- fillRandom(int n): Puts n tents randomly on the board.