

# User Story 106

## Description

“As a user, I want to be able to see the solution to the puzzle so that I can see if my answers were correct”

## Requirements

- The user should be able to see the answer to the puzzle at any time via a button press. Once the button is pressed the grid will be filled with the correct numbers

## Unit Test Cases

- Call the function used to solve the puzzle by passing in an unsolved version of the puzzle and check if every input after solving matches with the solution array and return true or false.

## Files

test2.js

userStory103.js

## Code

```
// Unit test to make sure the solution displayed matches with the solution matrix
it ('Correctly shows the solution to the sudoku puzzle', () => {
  assert.equal(operations.solve(userMatrix), true);
})
```

Figure 1: Unit test in test2.js for user story 106

```
userMatrix = [
  [0,0,0,8,0,0,9,6,0],
  [1,0,0,0,0,0,0,0,8],
  [0,0,0,3,5,0,0,0,0],
  [0,0,3,0,0,0,0,0,0],
  [0,0,0,0,0,0,0,0,7],
  [0,0,0,0,0,3,0,0,1],
  [0,2,1,0,0,4,8,9,6],
  [8,3,7,0,0,0,0,4,2],
  [0,0,4,0,0,0,0,1,0],
]
```

Figure 2: Matrix used to represent the 9x9 display of the grid on hard mode

```

const solve = (userMatrix) =>{

  matrixSolved = [
    [3,4,2,8,1,7,9,6,5],
    [1,5,6,2,4,9,3,7,8],
    [7,8,9,3,5,6,1,2,4],
    [2,1,3,4,7,5,6,8,9],
    [4,6,5,1,9,8,2,3,7],
    [9,7,8,6,2,3,4,5,1],
    [5,2,1,7,3,4,8,9,6],
    [8,3,7,9,6,1,5,4,2],
    [6,9,4,5,8,2,7,1,3],
  ]

  // set every index value equal to the solved matrix index value
  for(var i = 0; i < 9; i++){
    for(var j = 0; j < 9; j++){
      userMatrix[i][j] = matrixSolved[i][j]
    }
  }

  // Check if every value in the userMatrix is the same as the solution matrix
  for(var i = 0; i < 9; i++){
    for (var j = 0; j < 9; j++){
      if(userMatrix[i][j] !== matrixSolved[i][j]){
        return false;
      }
    }
  }

  return true;
}

module.exports = { solve }

```

Figure 3: Code used in userStory106.js to solve the userMatrix and check every number in the array

## ***Explanation***

The unit test will pass in a 2D array that represents the grid displayed in hard mode. Once the array is passed the function will take every index of the array and change each value to the solution value. Once the user matrix is filled with correct solution we check to make sure that in fact every value of the user matrix is equal to the solution matrix. If every value is equal we return back true but, if one value is not equal we return back false

## ***Result***

```
100659072@UOSL18-DVRT4H2 MINGW64 ~/Projects/testcases (master)
$ npm test

> testcases@1.0.0 test C:\Users\100659072\Projects\testcases
> mocha

  ✓ Correctly shows the solution to the sudoku puzzle

  1 passing (5ms)
```