# Data Section

## Variables

**Int size** –the first data from the test.txt file,forms the size of arrays.

**Int solved** – if true puzzle is solved.

**Int myArray[size][size]** –puzzle array to be solved. Data received from test.txt.

**Int copyArray[size][size]** – copy of myArray, used for more difficult puzzles.

**Int rowTotal[size]** – stores row totals.

**Int colTotal[size]** – stores column totals.

## Functions

(1)

**Int getSize()**

Gets size read in from test.txt with **fscanf()**.

(2)

**void fillArray(int \*\*array, int \*rowTotal, int \*colTotal, int count)**

Fills arrays.

(3)

**void createCopy(int \*\*myArray, int \*\*copyArray, int size)**

Creates copy of myArray if and when needed.

(4)

**void printUnsolvedArray(int \*\*array, int \*rowTotal, int \*colTotal, int size)** prints unsolved array.

(5)

**int checkTotals(int \*\*copyArray, int \*rowTotal, int \*colTotal, int size)**

Checks row and column totals, returns 1 if true.

(6)

**int solveArray(int \*\*myArray, int \*rowTotal, int \*colTotal, int size)**

Solves arrays with simple algorithm.

(7)

**int specialCase(int \*\*myArray, int \*\*copyArray, int \*rowTotal, int \*colTotal, int size)**;

Solves more difficult puzzles with extended algorithm.

(8)

**void printSolvedArray(int \*\*array, int size)**

Prints solved array to console.

Firstly **malloc()** creates memory space for arrays listed below.