

- 4 A computer program can generate a simple Sudoku puzzle using a 4×4 two-dimensional array.

An example of this puzzle is:

4	3	2	1
1	2	4	3
3	4	1	2
2	1	3	4

The first step to creating this puzzle is to develop a program to display the 4×4 two-dimensional array as a grid. This program will display the grid as:

```

4 3 2 1
1 2 4 3
3 4 1 2
2 1 3 4

```

Task 4.1

Create a program design that will declare, initialise and display the example puzzle shown. This design will:

- make use of top-down design
- include the data structure to represent the puzzle as a grid
- initialise the grid using the values shown
- make use of appropriate procedures and/or functions.

Evidence 13

Your program design for Task 4.1.

[6]

Task 4.2

Write program code to display the puzzle designed in Task 4.1.

Evidence 14

Your program code.

[5]

Evidence 15

Screenshot of the displayed grid.

[1]

The puzzle is said to be valid if it follows these rules:

- It consists of four quadrants.
- The numbers in each quadrant must add up to ten.
- Each horizontal and vertical row of the puzzle must also add up to ten.
- No number can be repeated in the same row, same column or same quadrant of the puzzle.

A good strategy for creating puzzles is to start with a valid 'base' puzzle and perform transformations on it to create new puzzles.



You will write program code to create new valid puzzles.

Each puzzle created will have **two** randomly selected transformations, from a possible four, performed on it. The following are the four possible transformations that can be carried out.

Transformation	Explanation	
1	Swaps two rows in the same quadrants	
2	Swaps two columns in the same quadrants	
3	Swaps the top and bottom quadrant rows entirely	
4	Swaps the left and right quadrant columns entirely	



Task 4.3

Write additional program code, with brief **internal commentary** to identify each transformation.

The program code will:

- create a method of selecting, at random, two of the four possible transformations to be applied to the puzzle
- call a sub-program for each of the required transformations
- randomly select which rows will be transformed for transformations 1 and 2, for example, either the top or bottom two rows (for transformation 1) OR either the left-most or right-most two columns (for transformation 2) respectively
- display the puzzle before each transformation is applied and after the final transformation. Before each transformation, it will also display the name of the transformation being carried out. For example:

```
4321
1243
3412
2134
```

Transformation 1: Swaps two rows in the same quadrants

```
1243
4321
3412
2134
```

Transformation 4: Swaps the left and right quadrant columns entirely

```
4312
2143
1234
3421
```

Evidence 16

Your program code that includes **internal commentary**.

[14]

Evidence 17

Screenshots of the output that shows each of the four transformations applied.

[4]

