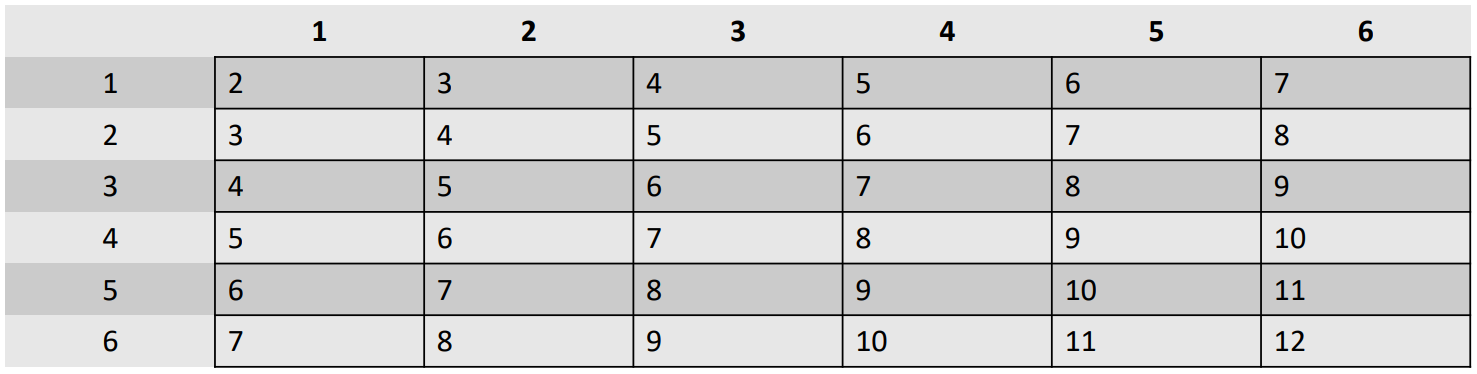
1. **(Dice Rolling)** Write a program that simulates the rolling of two dice. The program should use rand twice to roll the first die and second die, respectively. The sum of two values should then be calculated. [Note: Because each die can show an integer value from 1 to 6, then the sum of the two values will vary from 2 to 12, with 7 being the most frequent sum and 2 and 12 the least frequent sums.] The following figure shows the 36 possible combinations of the two dice. Your program should roll the two dice 36000 times. Use a one-dimensional array to tally the numbers of times each possible sum appears. Print the results in a tabular format. Also, determine if the totals are reasonable; ie., there are six ways to roll a 7, so approximately one-sixth of all the rools should be 7.



1. **(Turtle Graphics)** The Logo language made the concept of turtle graphics famous. Imagine a mechanical turtle that walks around the room under the control of a C program. The turtle holds a pen in one of two positions, up or down. While the pen is down, the turtle traces out shapes as it moves; while the pen is up, the turtle moves about freely without writing anything. In this problem you’ll simulate the operation of the turtle and create a computerized sketchpad as well.

Use a 50-by-50 array floor which is initialized to zeros. Read commands from an array that contains them. Keep track of the current turtle position at all times and whether the pen is currently up or down. Assume that turtle always starts at position 0, 0 of the floor with its pen up. The set of turtle commands your program must process are shown in the table below. Suppose that the turtle is somewhere near the center of the floor. The following “program” would draw and print a 12-by-12 square:

2

5,12

3

5,12

3

5,12

3

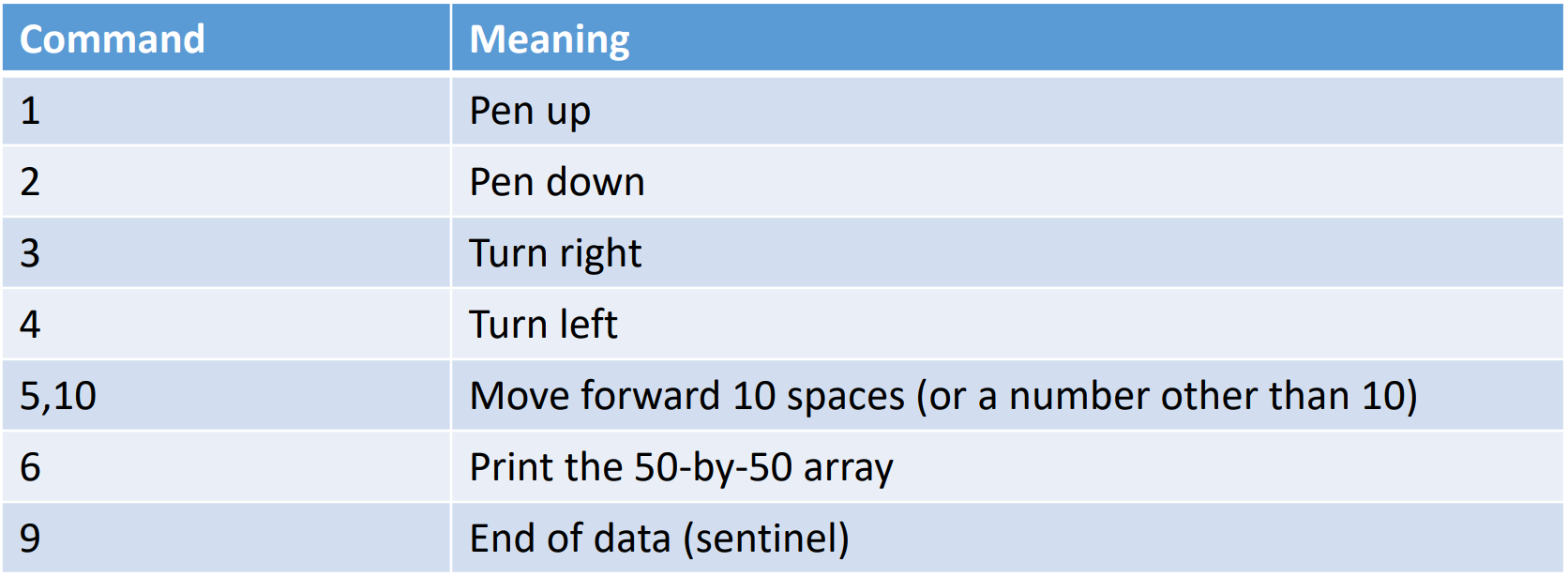
5,12

1

6

9

As the turtle moves with the pen down, set the appropriate elements of array floor to 1s. When the 6 command (print) is given, wherever there’s a 1 in the array, display an asterisk. Wherever there’s a zero, display a blank. Write a program to implement the turtle graphics capabilities discussed here. Write several turtle graphics programs to draw interesting shapes.



1. (Play a tic-tac-toe game) In a game of tic-tac-toe, two players take turns marking an available cell in a 3×3 grid with their respective tokens (either X or O). When one player has placed three tokens in a horizontal, vertical, or diagonal row on the grid, the game is over and that player has won. A draw (no winner) occurs when all the cells on the grid have been filled with tokens and neither player has achieved a win. Create a program for playing a tic-tac-toe game.

The program prompts two players to alternately enter an X token and O token. Whenever a token is entered, the program redisplays the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:

---------------------

| | | |

---------------------

| | | |

---------------------

| | | |

---------------------

Enter a row (0, 1, or 2) for player X: 1

Enter a column (0, 1, or 2) for player X: 1

---------------------

| | | |

---------------------

| | X | |

---------------------

| | | |

---------------------

Enter a row (0, 1, or 2) for player O: 1

Enter a column (0, 1, or 2) for player O: 2

---------------------

| | | |

---------------------

| | X | O |

---------------------

| | | |

---------------------

…..

….

---------------------

| X | | |

---------------------

| | X | O |

---------------------

| | | X |

---------------------

X player won.