APCS

Unit 4 Programming Project

1. For each given area code the phone company assigns a customer a 7-digit telephone number. Suppose we needed to store this data in a list, but we wanted to make sure there were no duplicate numbers entered. Write a program that will allow the user to enter as many numbers as they wish, but if they enter a number that is already in the list, the program will prompt them for a new entry. When the user is finished, the program should display the final list. You may assume they will always enter a 7-digit, positive integer. You may also assume they will never enter more than 100 numbers.

2. A "Magic Square" is a 4 X 4, 2D array of values where each row, column, and diagonal have the same sum.

1	14	14	4
11	7	6	9
8	10	10	5
13	2	3	15

In this case, the sum of all rows, columns and diagonals is 33.

Write a program that will display the array to the user in table form and tell them if the table makes a "Magic Square" or not. Try to make the program efficient, in that, once it knows the square is not 'magic', it should not continue to look at other rows/columns/diagonals.

3. Write a program that plays a simplified version of the card game “War”. Deal half of the cards to the player, half to the computer. Instead of taking cards, this game will keep score. On each ‘turn’ each player will turn over a card, the player that has the higher card will get the difference in their values as points. When all cards are played, the player with the highest total wins. The point values are as follows:

A: 1 point

2 – 10: Face Value

J: 11 points

Q: 12 points

K: 13 points

Each round should be initiated by the user. In the event of a tie, no points are awarded.

The final score should be displayed.

4. You will write a program that will display an 8 X 8 grid. Some of the cells will be 'occupied'. Each cell should have a 40% chance of being occupied when you fill in the grid. The program should then find any cell that is 'isolated'. This means that the cell is occupied, but has no neighbor to the north, south, east, or west. The program should count and display the total number of isolated cells.

See an example on the next page.

```
Here is a map to show the occupied cells in the grid.

_ * _ * * _ * _

* _ * * _ * _ _

* _ _ _ * _ * *

_ * _ _ * * _ _

* _ * _ * * _ _

_ * _ * _ _ _ *

_ * * _ * _ _ _

_ _ _ _ _ _ * *


0 1

0 6

1 5

3 1

4 0

4 2

5 3

5 7

6 4

9 cells are isolated.
```