Programming Assignment 4

1. Write code to generate a list of all even numbers, called **allEven**, that are divisible by 3 between 3 and 1000 (5 points)

2. Write code to generate a list of the first 10 prime numbers, called **primeNum**. (Remember that prime numbers are only divisible by 1 and itself). (5 points)

3. Write a function, called **firstLast**, that takes a list as a parameter and returns a list of the first and last elements in the list. (5 points)

4. Write a function, called **listSum**, that takes a list of numbers as a parameter and returns the sum of the values in the list. (5 points)

5. Write a function, called **commonElems**, that takes two lists as parameters and returns a new list that contains the elements that are common between the two. Your result list should contain no duplicates.
   For example, if the two lists are [1, 5, 6, 5, 2] and [3, 5, 1, 9] then the result list is [1, 5]. Note that the list sizes don't have to be equal. (10 points)

6. Write a function, called **uniqueElems**, that takes a list as a parameter and returns all unique elements in the list (i.e elements that appear only once).
   For example, if the input list is [1, 1, 1, 1, 1, 2, 3, 4] then the function should return [2, 3, 4]. (10 points)

7. Write a function, called **sortList**, that takes a list of numbers and returns a new list with the elements in the list appearing in sorted order from smallest to largest. Do not use the built-in sort or sorted functions. (10 points)

8. Write a function, called **rotateList**, that takes a list **a** and an integer **k** and rotates the list k spots to the left with wrap around.
   For example: if the list is [1, 4, 7, 13, 9] and k = 2 then the list is shifted three spots to the left with wrap around so the list becomes: [7, 13, 9, 1, 4] (10 points)

9. You are asked to implement “Last Match” which is a two player game. The game is played as follows: (40 points)

You start out with 5 bins of matches, and each bin contains 7 matches.

At each player’s turn, the player has to remove a number of matches from one of the bins. The player can choose the bin and the number of matches removed from the bin.

The last player to remove a match loses the game.

What to do:

■ Create a list of 5 integers to represent the bins of matches and initialize every element of the list to 7.

- Create a function named **removeMatches** that takes as parameters: the bin number, the number of matches to remove, and the list of bins. The function removes the number of matches requested from the requested bin. The function **returns True** if the matches were removed successfully and **False otherwise**. (False if there are not enough matches in the bin)

- Create a function named **checkGameEnd** that takes as parameters: the list of bins. The function checks if the bins are all empty and if so it **returns True**, otherwise it **returns False**.

- Create a function named **displayGame** that takes as parameters: the list of bins. The method will display the game bins.

- Use the functions you wrote to create the game for two players.

Hints:

- Note that you will need a way to determine whose turn it is to remove matches. Keep this simple so a variable is sufficient.

- **Use the return value of the function removeMatches** to determine if the move the player made is legal (it's not legal if they are removing a number less than or equals to 0, or if they remove more than the number of matches in the bin). If the move is not legal then keep prompting the player to re-enter the bin and the match count until they give a legal move.

- **Use the return value of the function checkGameEnd** to determine the end of the game.
- Check that the player has picked an appropriate bin number as well (there are only 5 bins, so if the user enters 6932, it should prompt him/her to pick again).

Sample Run:

```
The Bins:

 7          7        7        7       7
|__|      |__|     |__|     |__|    |__|

Player One's Turn:  choose a bin 2

Player One's Turn: How many matches are you removing? 5

The Bins:

 7          2        7         7       7
|__|      |__|     |__|     |__|    |__|

Player Tow's Turn:  From which bin are you removing matches? 4

Player Two's Turn: How many matches are you removing? 3
```

```
The Bins:

 7         2        7         4       7
|__|      |__|     |__|      |__|    |__|
```

… the game continues until

```
The Bins:
 0         1        0          0       0
|__|      |__|     |__|      |__|    |__|

Player Tow's Turn:  From which bin are you removing matches? 2

Player Two's Turn: How many matches are you removing? 1

The Bins:
 0         0         0         0       0
|__|      |__|     |__|      |__|    |__|

Player One Wins!!!
```