Exercises Week 2: Flow Controls & Loops

General Note: This set of exercises is designed to give you a serious boost in ability. These exercises are complex and will require good logic methodology as well as good coding. You should do your own research if you don't understand the algorithms exactly. It is not easy to explain them without visual aid. Since these are not required for anything, you can obviously look up solution code. However, if you do that I would recommend that you understand how the solution works too. Kudos to whoever manages to solve all four without any external code.

Exercise 1: Calculator (Basic)
Using a switch statement create an "advanced" calculator that allows the user to Add, Subtract, Multiply and Divide. For the sake of simplicity have the user enter the numbers before they chose what operation they want. The output should be the outcome of the numeric operation.

Exercise 2: Binary Search (Intermediate)
Using a the following array, create a binary search algorithm. A binary search algorithm is one that works on a sorted list and what it does is: It finds the middle of the array and checks if the value of the query is smaller or larger. If it is smaller it then "cuts" the array in half and does the same thing only now the array is smaller than the middle array. Similarly if it was larger then the array changes to the one larger than half. This process continues until the element is found.

Search for 47

0	4	7	10	14	23	45	47	53

This is a bit more complex and you should try to write it on your own but it is imperative that you understand what this algorithm executes. Research it first, deconstruct the logic and code it.
The array you should use:

Array1 = [0,2,4,6,8,10,12,14,16,18,20,22,24]

The query should be a user input. The output should be either: the query exists or, the query does not exist

There is more in Page 2.

Exercise 3: Bubble Sort (Advanced)
Using the array bellow, sort the array by comparing adjacent elements. If the elements are in the wrong order then swap them. We want the array to end up in ascending order (smallest to largest). Remember that you will need to run the compare & switch multiple times through the array.

6 5 3 1 8 7 2 4

The Array that you should use is:

array1 = [12,24,1,9,42,8,3,0]

The output should be a print of the sorted array. You can do this with a for loop.