Week 3: Arrays, Lists & Dictionaries

Exercise 1: Say when

Level: Easy
Create an empty array and prompt the user up until how many elements he wants the array to hold. Fill this array with the Arithmetic sequence: ***a_n = 2 + 2(n-1)*** *(2,4,6,8,10,...)*. Up until the number of elements that the user inputted.

I.e. if the user allowed 100 elements, the array should have a len() = 99 and the last element should be 198.

Focus: Array appending

Exercise 2: The one that doesn’t fit in

Level: Easy
Using the *list* defined below, which will be of general type **int** find the element or elements that are not of this type and remove them from the list.

list = [1, 2, 3, 4, 5, '6', 7, 8, 9, 10]

Focus: Type checking

Exercise 3: Array Translation

Level: Easy
Using a list and a dictionary, convert a set of elements that don’t fit using the dictionary. Below are the list & dict.

list = ["VB", "V0-", "V0", "V0+", "V1", "V2", "V3", "V4", "V5", "V6", "V7", "V8", "V9", "V10"]
dict = { "VB" : 1, "V0-" : 2, "V0" : 3, "V0+" : 4, "V1" : 5, "V2" : 6, "V3" : 7, "V4" : 8, "V5" : 9, "V6" : 10, "V7" : 11, "V8" : 12, "V9" : 13, "V10" : 14
}

Focus: Lookup Dictionaries

Exercise 4: We don’t like John

Level: Easy
Given a set of parallel (linked) arrays you need to find and delete the information pertaining to someone named John but you need to also *print* John's time before you delete it. Ensure a correct use of pop(), remove() and index retention. The arrays are as follows:

```
names = ["mike", "tony", "greg", "shaq", "elias", "john"]
times = [10.2, 11.3, 11.7, 10.1, 14.5, 13.2, 12.1]
```

Focus: Parallel Arrays