Solutions to Assignment 1

Author: Pooya Aghanoury

This is a simple module composed of `main.py` as well as an `assets/` directory containing extra files necessary for the completion of this assignment. Run `main.py` via python3, which prints out the results for the first exercise. It will then prompt the user to enter a string to determine if the char array 'abcd' is found within it.

Note: See end of document for entire source code

Questions from Handout

1. An IDE stands for Integrated Development Environment. Used to develop software, an IDE also offers unique and custom features for the benefit of the user's workflow
2. Exceptions can be handled with `try/except` statements. Example shown below

```
try:
    "try some operation"
except:
    "do this if it throws an error"
```

3. Define a function with

```
def some_function(arg1, arg2,...argN):
    "Do someting"
    return
```

4. `True`. Functions are defined with a set of inputs. Though, some of these inputs can be optional.

5. Two methods used on list objects are the `append()` method, for appending elements to a list. There is also a `reversed()` method, which returns essentially a reversed version of a list.

6. Implementing the state machine with pointers allows more states to be dynamically added or removed. While the if statements require hard coding states and conditions, not ideal for dynamic situations.

Exercise 1

1. Maximum Number: **99**
2. Minimum Number: **2**
3. Index of '38': **76**
4. Most Frequent Element: **76**
5. A sorted list: [2 2 3 4 5 6 6 6 7 8 10 10 10 10 11 12 13 16 16 17 17 18 19 20 20 23 26 27 27 27 29 29 32 32 33 33 33 34 34 34 35 36 38 39 41 42 44 46 46 47 48 49 49 50 52 54 55 55 56 59 63 63 64 65 67 67 68 68 69 69 71 72 73 73 73 73 73 74 75 75 75 76 79 79 80 81 81 82 83 83 85 86 86 87 88 89 89 94 95 99]

6. Even Numbers: [2 4 6 8 10 12 16 18 20 26 32 34 36 38 42 44 46 48 50 52 54 56 64 68 72 74 76 80 82 86 88 94]

Exercise 2

Run `main.py` to test this module.

Source Code

```
import numpy as np
"""this module solves the problems layed out in exersize 1 and 2 """
# read text file
with open('assets/datafile.txt','r') as f:
    b = eval(f.read())

# brute force approach to finding the most frequent number
max_val = 0
most_frequent = None
for i in b:
    if b.count(i) > max_val:
        max_val = b.count(i)
        most_frequent = i

# find all of the even numbers
even_numbers = [i for i in set(b) if i % 2 == 0]

# print results
print("Maximum from list:", max(b))
print("Minimum from list:", min(b))
print("Index where '38' is located:", b.index(38))
print("The most frequently occuring:",most_frequent)
print("A sorted list:", np.sort(b, axis=0))
print("Even Numbers:", even_numbers)


# this module solves the problems layed out in exersize 2
# a crude attempt at a state machine


"""Begin excersize 2"""
print("\nBeginging assignment 2")
string = input("Enter a string to find 'abcd': ")
# in python, we can search if the string contains 'abcd'
# using the string.find('abcd'), shown below as well

# if string.find('abcd') != -1: print("found sequence 'abcd' (python-
style)") # this returns the
# else: print("did not find sequence 'abcd' (python-style)")

# but lets also do it the proper state machine style
state = 0
for i in string:
    if i == 'a': # regardles of current state, if we find 'a' go to state
1
        state = 1
    elif i == 'b' and state == 1:
        state = 2
    elif i == 'c' and state == 2:
        state = 3
    elif i == 'd' and state == 3:
        state = 0
        print("Found sequence 'abcd' (state machine)")
        exit(0)
    else:
        state = 0
# if we get to this point, we did not find it
print('did not find sequence (state machine)')
```