1)

A) What is an IDE?

IDE stands for Integrated Development Environment. IDE’s bring together many different aspects of writing a computer program. IDE’s allow for editing code, syntax highlighting, autocomplete, building executables, and debugging [1].

B) What are the key words to handle exceptions in Python?

When dealing with exceptions in python one can use the key words “try” and “except”. One can use multiple except statements with different types of error like IndexError and ValueError.

C) How do you define a function in Python?

To define a function in python you need to put the def keyword in front of the Python’s function name. Then have parentheses with the parameters of the function. End the function declaration with a colon.

D) True or False - when calling a specific function, the number of arguments is fixed.

False- The function can have multiple parameters passed or none depending on how the function was created.

E) What are two methods that can be used with list objects, and what do those methods do?

Method 1) sort() – This will sort the list

Method 2) copy() – This will copy the list to a new list without pointing back to the old list.

F) Carefully describe all the differences between the example state machine implemented with if statements, and the example state machine implemented with function pointers. How do pointers make it easier to add new states?

The main difference between the example state machine with if statements and function pointers is the different variable tracking the current state. In the if statements case a int is tracking the case while in the pointer example the name of the state is the tracker. I find it easier to follow if you use the name of the state to track. Pointers make it easier because very little code needs to be implemented in your main function and all the activity happens in each of the states. The if statement case needs to call every different state depending on the state variable. This can be tedious and hard to track.

[1] https://www.codecademy.com/article/what-is-an-ide

2) Commented Code.

1)

'''

AUTHOR: Mark McNulty

PURPOSE: The purpose of this program is to take a list of numbers and return modifications of that list to the user

USAGE:   python3 ExerciseOne.py

         python ExerciseOne.py

'''

# define the repeats function

def repeats(inputList):

    # make a new list with zeros in every element

    repeat = [0]\*100

    # loop through the input list

    # increment the value of the element in the repeat list that corresponds to the value in the input list

    for i in inputList:

        repeat[i] += 1

    return repeat

# define even numbers in order

def allEvenNumbersInOrder(inputList):

    # sort the list

    inputList.sort()

    # make a new list

    outputList = []

    for i in range(len(inputList)):

        # if the element is even, add it to the output list

        if inputList[i] % 2 == 0:

            outputList.append(inputList[i])

    return outputList

# read in the input file

with open('datafile.txt', 'r') as file:

    data = eval(file.read())

# print the max value in the list

print("max:", str(max(data)))

print()

# print the min value in the list

print("min:", str(min(data)))

print()

# print the index of 38

print("index of 38:", str(data.index(38)))

print()

# print the number of times each number is repeated

# this calls the function repeats

print("number of times each number is repeated: ", end="")

print(repeats(data))

print()

# make a new copy of the list

sortedList = data.copy()

# sort the list

sortedList.sort()

# display the list

print("sorted list:", end="")

print(sortedList)

print()

# make copy of the list

allEven = data.copy()

# print out all even in order

print("all even numbers in order:", end="")

# call the function

print(allEvenNumbersInOrder(allEven))

2)

'''

AUTHOR: Mark McNulty

PURPOSE: This program will take a string as an input and display if the string contains the 'abcd' sequence.

         This is done using a state machine.

USAGE:   python3 ExerciseTwo.py

         python ExerciseTwo.py

'''

# define the states

# define state0

def state0(input):

    # move to stateA if input is a

    if input != 'a':

        return state0

    else:

        return stateA

def stateA(input):

    # move to stateB if input is b

    # stay in stateA if input is a

    # return to state0 if input is not a or b

    if input == "a":

        return stateA

    elif input == "b":

        return stateB

    else:

        return state0

def stateB(input):

    # move to stateC if input is c

    # go back to stateA if input is a

    # return to state0 if input is not a or c

    if input == "a":

        return stateA

    elif input == "c":

        return stateC

    else:

        return state0

def stateC(input):

    # move to stateD if input is d

    # go back to stateA if input is a

    # return to state0 if input is not a or d

    if input == 'a':

        return stateA

    elif input == 'd':

        stateD(input)

        return state0

    else:

        return state0

def stateD(input):

    # print that the string contains the sequence

    print("abcd is contained in the string")

    return None

# create dictionary to describe the state machine

state\_dictionary = {

    state0: "No A yet",

    stateA: "state A",

    stateB: "State B",

    stateC: "State C",

    stateD: "State D"

}

# create main loop

if \_\_name\_\_ == "\_\_main\_\_":

    # get the string from the user

    input\_string = input("Enter a string: ")

    # make pointer to state 0

    state = state0

    # loop through the string

    i = 0

    while state is not None:

        # call the state function

        new\_state = state(input\_string[i])

        # move to the next state

        state = new\_state

        # break if the end of the string is reached

        if i == len(input\_string) - 1:

            break

        # increment the counter

        i += 1

    print("done with state machine")