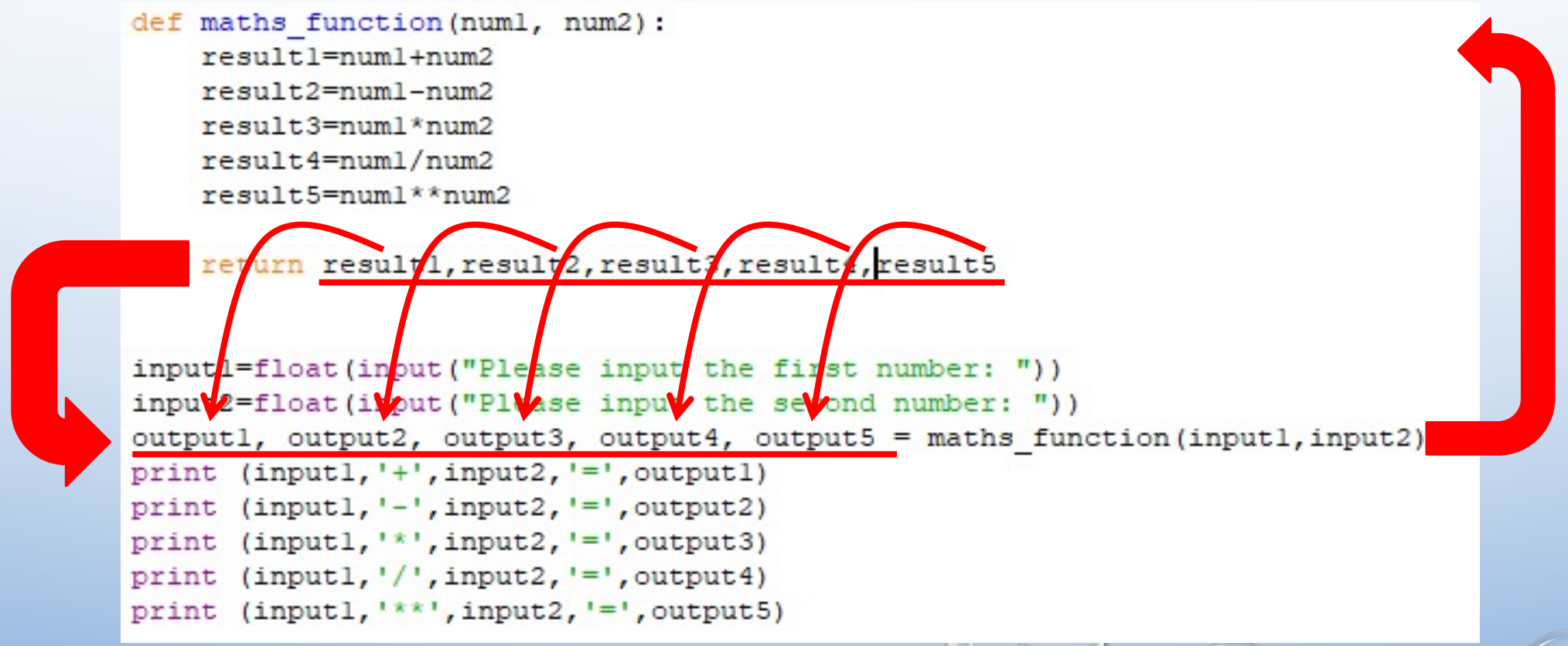
KIT100 Programming Preparation

Tutorial Eight – Week 9

Today's Flow

- Walk through Portfolio Tasks CR9.1, CR9.2 and CR9.3.
- One-on-one session with me
  - to help you to resolve the programming concerns from the previous weeks

Functions


def maths_function(num1, num2):
result1=num1+num2
result2=num1-num2
result3=num1*num2
result4=num1/num2
result5=num1**num2
return result1,result2,result3,result4,result5
input1=float(input("Please input the first number: "))
input2=float(input("Please input the second number: "))
output1, output2, output3, output4, output5 = maths_function(input1,input2)
print (input1,'+',input2,'=',output1)
print (input1,'-',input2,'=',output2)
print (input1,'*',input2,'=',output3)
print (input1,'/',input2,'=',output4)
print (input1,'**',input2,'=',output5)

Python Lists

- List is a collection which is ordered and changeable. Allows duplicate members.
- Access List members by using index 0,1,2,…,n
- Python provides different functions to manipulate list
  - insert
  - append
  - pop
  - remove

Python List Methods
append() - Add an element to the end of the list
extend() - Add all elements of a list to the another list
insert() - Insert an item at the defined index
remove() - Removes an item from the list
pop() - Removes and returns an element at the given index
clear() - Removes all items from the list
index() - Returns the index of the first matched item
count() - Returns the count of the number of items passed as an argument
sort() - Sort items in a list in ascending order
reverse() - Reverse the order of items in the list
copy() - Returns a shallow copy of the list

Python Lists

- Basic list functions
  - Create a list
  - Print a list
  - Access elements from a list
  - Loop through a list
  - Add/Append an element
  - pop an element
  - Count number of elements from a list

```
1  a = [1,2,3,4,5]
2
3  print("Original list: ",a)
4  print("0th element of list: ",a[0])
5
6  print("Loop all elements from list")
7  for ele in a:
8      print("Element:",ele)
9
10 a.insert(0,6)
11 print("After insert() 6: ",a)
12 a.append(7)
13 print("After append() 7: ",a)
14 a.pop()
15 print("After pop() : ",a)
16 print("Number of element of list", len(a))
```

Python Lists

```
1  a = [1,2,3,4,5]
2
3  print("Original list: ",a)
4  print("0th element of list: ",a[0])
5
6  print("Loop all elements from list")
7  for ele in a:
8      print("Element:",ele)
9
10 a.insert(0,6)
11 print("After insert() 6: ",a)
12 a.append(7)
13 print("After append() 7: ",a)
14 a.pop()
15 print("After pop() : ",a)
16 print("Number of element of list", len(a))
```

Output:

```
Original list:  [1, 2, 3, 4, 5]
0th element of list:  1
Loop all elements from list
Element: 1
Element: 2
Element: 3
Element: 4
Element: 5
After insert() 6:  [6, 1, 2, 3, 4, 5]
After append() 7:  [6, 1, 2, 3, 4, 5, 7]
After pop() :  [6, 1, 2, 3, 4, 5]
Number of element of list 6
```

Let’s guess the output

```
1  def give_me_a_list(i_am_a_list):
2      while len(i_am_a_list) > 2:
3          i_am_a_list.pop()
4
5  cheat_code=["how", "do", "you", "turn", "this", "on"]
6  give_me_a_list(cheat_code)
7  print(cheat_code)
```

Output will be …

A) ['how', 'do', 'you', 'turn', 'this', 'on']
B) ['how', 'do’]
C) ['how', 'do', 'you’]
D) ['this', 'on']

Let’s guess the output

```
1  def give_me_a_list(i_am_a_list):
2      while len(i_am_a_list) > 2:
3          i_am_a_list.pop()
4
5  cheat_code=["how", "do", "you", "turn", "this", "on"]
6  give_me_a_list(cheat_code)
7  print(cheat_code)
```

Output will be …

A) ['how', 'do', 'you', 'turn', 'this', 'on']
B) ['how', 'do’]
C) ['how', 'do', 'you’]
D) ['this', 'on']

Let’s guess the output

```
1  hello_world=["You","know","the","rules","and","so","do","I"]
2  another_list=hello_world
3
4  another_list[0] = "I"
5  print(hello_world)
```

Output will be …

A) I
B) [‘You', 'know', 'the', 'rules', 'and', 'so', 'do', 'I']
C) ['I', 'know', 'the', 'rules', 'and', 'so', 'do', 'I’]
D) You

Let’s guess the output

```
1  hello_world=["You","know","the","rules","and","so","do","I"]
2  another_list=hello_world
3
4  another_list[0] = "I"
5  print(hello_world)
```

Output will be …

A) I
B) ['You', 'know', 'the', 'rules', 'and', 'so', 'do', 'I']
C) ['I', 'know', 'the', 'rules', 'and', 'so', 'do', 'I']
D) You

In this program, we have provided a **reference** to the hello_world with another name another_list but these two lists are same which have two references(hello_world and another_list). So any alteration with another_list will affect the original list (hello_world).

Let's guess the output

```
1  list_1=["I","am","2",1]
2  for ele in list_1:
3      print(ele*2)
```

Output will be …

A) II
amam
22
2

B) I
am
2
1

C) ["I","am","2", "1"]

D) Syntax Error

Let's guess the output

```
1  list_1=["I","am","2",1]
2  for ele in list_1:
3      print(ele*2)
```

Python allows multiple data types in a list.
But you have to be careful while loop through the list.

Output will be …

A) II
amam
22
2

B) I
am
2
1

C) ["I","am","2", "1"]

D) Syntax Error

Portfolio Task

9.1CR Lists

Description:

The pharmacist from task 8.1PP was so impressed by your work they recommended you to their construction-builder friend. The builder has to assign work to two groups of tradesman subcontractors on a fair basis. However, their ancient, 1980s work-allocation program only gives them entire lists of numbers that corresponds to each worker. The builder wants the lists to be broken down into the two groups - one with the odd number workers and one with the even number workers, so he can then allocate work to each group. He also needs to know how many people are in each group.

Task:

Write a program that defines and uses a **function** called `displayAndCountOddsAndEvens` that takes one named parameter `numbers` which will be a list.

The function should display (i.e print to the output) the contents of the `numbers` list, and then:

- The title: "Odd numbers:" and then all odd numbers in the `numbers` list (one perl line) - but only if there are one or more odd numbers
- The title: "Even numbers:" and then all even numbers in the `numbers` list (one perl line) - but only if there are one or more even numbers

The function should then return two items - the count of how many odd numbers there were, as well as how many even numbers there were - **note the function does not print out these counts**.

Test Data:

In the main code, you **must** test your function with the following lists - it is ok to alternately define the list and call the function several times for this task.

```
list = [1,2,3,4,5,6,7,8,9,10]
# call your function and print the count results
list = [2,4,6,8,10]
# call your function and print the count results
list = [1,3,5,7,9]
# call your function and print the count results
list = []
# call your function and print the count results
```

Example output:

```
The list is: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Odd numbers:
1
3
5
7
9
Even numbers:
2
4
6
8
10
There were 5 odds and 5 evens

The list is: [2, 4, 6, 8]
Even numbers:
2
4
6
8
There were 0 odds and 4 evens
```

9.2 CR – List Operations

Task:

Submit a ***plain text*** document (i.e. use Notepad or TextEdit) that contains the answers to the following questions:

1. Given `lst = [30, 1, 2, 1, 0]`, what does each of the following statements do and what is the contents of `lst` after applying **each** of the following statements? *Assume that each line of code is independent (i.e. start with the original list each time).*

1a	`lst.append(40)`
1b	`lst.insert(1, 43)`
1c	`lst.extend([1, 43])`
1d	`lst.remove(1)`
1e	`lst.pop(1)`
1f	`lst.pop()`
1g	`lst.sort()`
1h	`lst.reverse()`

2. Given `lst = [30, 1, 2, 1, 0]`, what does each statement do and what is the **return value** of each statement? *Assume that each line of code is independent.*

2a	`lst.index(1)`
2b	`lst.count(1)`
2c	`len(lst)`
2d	`lst[1 : 3]`
2e	`lst[3]`

Hint:

Your answers should detail both what the string method does and what the result is.

For example:

1a: append(40) making the number 40 will be added to the end of the list. `lst = [30,1,2,1,0,40]`.

Submission Details

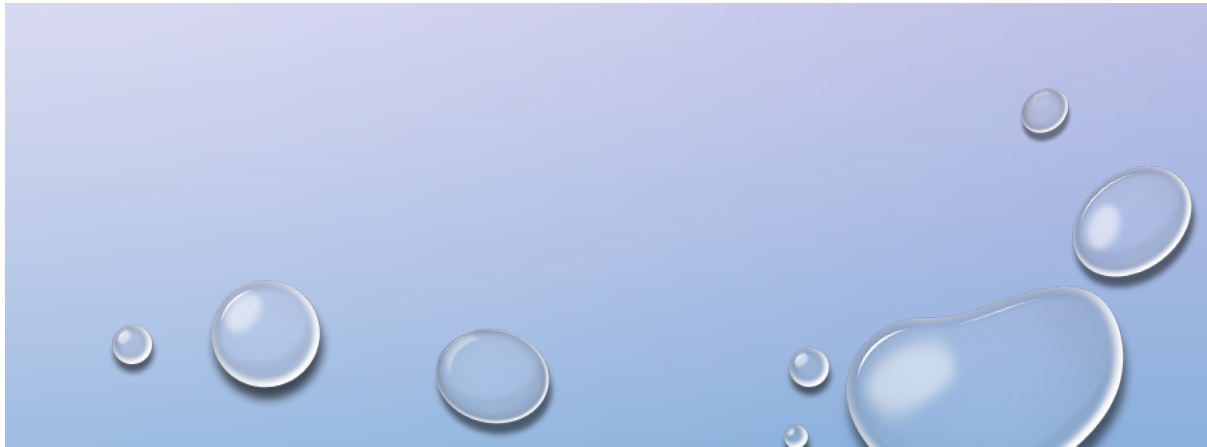
Upload the following to the MyLO submission folder for this task:

1. The text file (i.e. the a plain text file containing your answers)

Assessment Criteria & Hints

A completed submission **must**:

1. Give correct answers for all questions



9.3CR – Error Identification

Description:

You've been given some code that doesn't work. NASA was testing rockets and a certain number of them fail, and the code is supposed to allow an operator to enter rocket numbers that fail. The output would then indicate which rockets fail.

Task:

Your task is to find the errors in the code below - the errors might be syntax errors or runtime errors (for example, an undefined variable error is an example of a runtime error as it only occurs when the code tries to refer to the variable's value). You do not have to type the program in or get it running, you only have to indicate where the errors are. **There are eight errors to find.**

In a plain text document (e.g. notepad or textedit) , for each error:

1. Identify the line number where the error occurs
2. Specify what the actual error is

Note that where an error occurs, there's only one error on that line of code

Example submission:

The following is an example of the format of the file you must submit.

```
Error Number    Line Number    Error Description
1               Line 1         missing ] at end of list

...etc
```

Code:

```
1 frequency = [0,0,0,0,0,0,0,0,0,0
2
3 number = int(input("Please enter a number between 1 and 10:))
4
5 while number != 0
6    frequency[numbers - 1] += 1
7        number = int(input("Please enter a number between 1 and 10:"))
8
9
10 print()
11 print("Determining the frequency of numbers entered:"
12
13 count = 1
14 for item in freqncy:
15    if item != 0
16       print("Number: %2d, frequency: %2d" % (count, item))
17    count += 1
```