MA1008 Introduction to Computational Thinking Quiz 2.
Answer all the questions in the spaces provided.
AY 2021/2022, Semester 2, Week 9

Your Name: __________________________________ Group: _____

Solutions in Red

There are possibly different solutions to a question. Hence, if a student's solution isn't the same as that given here, please verify if it is correct before deducting marks.

1. Given the list
   `L = ["az", "bazd", "abz", ["az", "ab"], ("azaz", "zaza")],`
   state what are printed by the following `for` loop? (10 marks)
   ```
   for s in L:
      print("az" in s, end = " ")
   ```
   ```
   True True False True False
   ```

2. i. Given a variable `c` which carries a single character, write a Python expression that determines if `c` is a letter, either upper or lower case, using the `ord()` function. (6 marks)

   ```
   ord("a") <= ord(c) <= ord("z") or ord("A") <= ord(c) <= ord("Z")
   ```

   ii. Given a variable c which carries a single character, write a Python expression that determines if `c` is a letter, either upper or lower case, without using the `ord()` function or any string method, such as `isupper()`, `islower()`, `isalpha()` etc. (6 marks)

   ```
   "a" <= c <= "z" or "A" <= c <= "Z"
   ```
   or c in "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

   Note: Do not use the ASCII values of the characters directly.

3. i. Create a dictionary called `Week` using the first two letters of the weekdays (such as "Mo", "Tu", etc.) as the keys and the corresponding days, "Monday" … "Sunday", as the values. (4 marks)

   ```
   Week = {"Mo":"Monday", "Tu":"Tuesday", "We":"Wednesday",\
           "Th":"Thursday", "Fr":"Friday", "Sa":"Saturday",\
           "Su":"Sunday"}
   ```

   ii. Using the dictionary, write a Python program that checks each day and prints `True` if it contains the letters `"s"` and `"u"` and not the letter `"n"`, in upper or lower case, and `False` otherwise. (6 marks)

   ```
   for w in Week:
      week = Week[w].lower()   # convert everything to lower case
      if "s" in week and "u" in week and "n" not in week:
         print(Week[w])
   ```

4. i. Given the list `L = ["apple", "pear", "apple", "apple", "orange", "pear"],`

write a list comprehension statement that creates a new list from L without duplicates. You would need to initialise the new list to an empty list first. (7 marks)

```
new = []
[new.append(item) for item in L if item not in new]
```

If student uses new.extend(item), give 4 marks.

ii. Write a list comprehension statement that creates a new list without the empty strings, empty tuples and empty lists from the following list: (7 marks)

```
L = ["", 12, (3, 4, 5), (), [6, 7], "", [], "89"]
```

Hint: All empty items have the Boolean value `False`. All other items are `True`.

```
New = [item for item in L if item]
```

5. The following function, `modifyL`, has three parameters: `L`, a list, and two integers `m` and `k`, with `k` defaulted to `-1`. It is designed to modify `L` according to this specification: if `k` is negative, `L` is duplicated `m` times, otherwise element `k` of `L` is assigned the value `m`. You may assume that `k`, when not negative, is within the length of `L`.

```
def modifyL(L, m, k = -1):
   if k < 0:
      L = L * m  # duplicate L m times
   else:
      L[k] = m   # change value of L[k] to m.
```

i. The following statements call `modifyL` twice. State the value of `L1` after each call. (4 marks)

```
L1 = [1, 2, 3, 4, 5]

modifyL(L1, 2)                              [1, 2, 3, 4, 5]

modifyL(L1, 10, 2)                          [1, 2, 10, 4, 5]
```

ii. Does the function perform according to the specification? If not, state how you would modify it so that it does (no need to rewrite the whole function). If necessary, provide the corresponding modifications to the function call statements as well. (8 marks)

No, it doesn't work as specified when `k` is negative because `L = L * m` makes `L` local and therefore is not visible outside the function. Hence `L1` remains unchanged upon the first call. To rectify, add return `L` at the end of the function outside the `if`…`else`… block. Also need to assign `L1` to the call to `modifyL` for both the calls, such as `L1 = modify(L1, 2)`.

6. The following program contains syntax errors.

```
def Fun1(v1, v2, V1 = 1, V2 = 'start'):
    temp1 = v1[V1:] + v1[V1::-1]
    temp2 = V2
    for i in v2:
        temp2 += i
    if V2 == 'start':
        Fun2(v3 = temp1, V3 = v1, v4 = len(temp2))
    else:
        Fun2(v3 = temp1, v4 = temp2)

Fun1([1,2,3,4,5,6,7], [4,5,6,7,8], 3, 6)   # Statement 1

def Fun2(v3, V3 = [], v4, V4 = 0):
```

```
temp3 = v3 + V3
if v4 >= len(temp3):
    v4 = len(temp3)
print(temp3[v4:V4:-2])
```

i. State what the errors are. You do not need to rewrite the program correctly. (4 marks)

Two errors, first is in the definition of Fun2: v4, a non-optional parameter, must come before V3 an optional parameter. The second is the call to Fun1, which invokes Fun2. This call must come after the definition of Fun2, not before.

ii. Assuming that appropriate corrections have been made, what is printed by Statement 1? (2 marks)

```
[1, 3, 7, 5]
```

iii. Assuming that appropriate corrections have been made, what are printed by the following statements? If an error occurs, state what the error is without providing correction. (2 marks each)

a. `Fun1([11,12,13], 'strings')` `[13, 11, 12]`

b. `Fun1('strings', 'strings', 1, 'strings')` Error in call to `Fun2`, in the statement `temp3 = v3 + V3`, `v3` is a string and `V3` is a list. Can't add.

7. Given the tuple `T = (9, 8, [7, 6], "54", (3, 2))`, state the values of the variables on the left of the following statements. Where there is an error, state what the error is and provide a correction, and state the value of the variable on the left based on your correction: (2 marks each)

i. `V1 = T[0] + T[2][1]` `15`

ii. `V2 = T[1] + T[3][1]` Error. Can't add integer and string. Either change `T[1]` to `str(T[1])` or `T[3][1]` to `int(T[3][1])`.

iii. `V3 = T[2] * T[4][0]` `[7, 6, 7, 6, 7, 6]`

iv. `V4 = T[4] + 1` Error. Cannot add tuple and integer. Change `1` to `(1,)`.

v. `V5 = T[2].extend(T[4])` None. The `extend()` method does not return a value. This is not explicitly an error as the program still runs, but student can correct it by removing the assignment `V5 =`.

vi. `V6 = T[4].append(T[2])` Error. `T[4]` is a tuple and tuple has no extend() method. Change to T[4] + (T[2],)

2 marks each for these sub questions is too little for including the correction. So, please give full mark if the student only write "Error" without providing the correction for the parts with error.

8. Given the same tuple as in the previous question, `T = (9, 8, [7, 6], "54", (3, 2))`, state which of the following statements would lead to error and which not. Explain your answers. You do not need to provide the correction. (2 marks each)

i. `T[0] = T[1]` Error. `Tuple T cannot be modified.`

ii. `T[2][0] = T[4]` `OK.` `T[2]` is a list and is mutable.

iii. `T[2][2] = 5` Error. `T[2]` has no index 2. Index out of range.

iii. `T[3] *= 2` Error. `T` is a tuple and is immutable.

iv. `T[4] = (2, 3)` Error. `T` is a tuple and is immutable.

9. The following three lists record five different car models, the distance each travelled and the volume of fuel consumed in the corresponding elements of the lists.

```
Car = ["BMW", "Kia", "Nissan", "Toyota", "Ferrari"]
Distance = [182.32, 205.18, 126.43, 385.91, 98.22]  # in km
Volume = [22.13, 15.95, 9.02, 31.68, 13.78]  # in litres
```

Using the three lists, write Python statements to print the fuel consumption rate (in km/litre) of the cars in this format:

```
BMW□□□□□□Kia□□□□□□Nissan□□□Toyota□□□Ferrari□□
8.24□□□□□12.86□□□□14.02□□□□12.18□□□□7.13□□□□□
```

Note: Consumption rate = distance/volume. □ denotes the space character. You may assume that the three lists exist and need not include them in your code. (10 marks)

```
for car in Car:
   print(f"{car:8s}", end = " ")  # print model without new line
print()  # add a new line
for k in range(len(Car)):
   print(f"{Distance[k]/Volume[k]:<8.2f}", end = " ")
```

Note: students could use the .format() method associated with strings.

<> <> <> END OF PAPER <> <> <>