

## MA1008 Introduction to Computational Thinking Quiz 2.

Answer all the questions in the spaces provided.

AY 2022/2023, Semester 1, Week 9

Your Name: \_\_\_\_\_

Group: \_\_\_\_\_

# Solutions

There can be multiple solutions to a question. So if a student offers a different solution, please check first before marking.

1. Answer the following questions given these assignments,

```
L1 = L2 = [1, (2, 3), 4]
```

```
T1, T2 = (4, [5, 6], 7), (4, [5, 6], 7)
```

Where there is an error, state why the error occurs. You do not need to provide corrections. All the questions use the above assignments and do not depend on the outcome of a previous question.

(2 marks each)

- i. What are the values of L1 and L2 after the statement `L1[1] = 2`?

`L1 = L2 = [1, 2, 4]`

- ii. What are the values of L1 and L2 after the statement `L2[1][0] = 2`?

Error, as `L2[1]` is a tuple and therefore immutable.

- iii. What are the values of L1 and L2 after the statement `L1 = L1.pop(1)`?

`L1 = (2, 3)`, `L2 = [1, 4]`

- iv. What are the values of T1 and T2 after the statement `T1[1] = 2`?

Error, as T1 is a tuple and hence immutable.

- v. What are the values of T1 and T2 after the statement `T2[1][0] = 2`?

`T1 = (4, [5, 6], 7)`, `T2 = (5, [2, 6], 7)`

- vi. What are the values of T1 and T2 after the statement `T1 = T1.pop(1)`?

Error as T1 is a tuple and tuple doesn't have the `pop()` method.

2. i. What is the content of the dictionary `fruit` upon executing the program below? (5 marks)

```
fruit = {"A": "Apple", "B": "Banana", "G": "Guava", "P": "Pear"}
fruit["P"] = "Peach"
del fruit["G"]
fruit[5] = "Healthy"
fruit = {"A": "Apple", "B": "Banana", "P": "Peach", 5: "Healthy"}
```

- ii. Following the above program, write a `for` loop that prints all the members of `fruit` that contains the letter "a" and not the letter "p", both of either case. (5 marks)

```
for key in fruit:
    if "a" in fruit[key].lower() and "p" not in fruit[key].lower():
        print(fruit[key])
```

3. i. What is produced by this list comprehension statement? (5 marks)  
`[x+y**2 for x in range(1, 5) for y in range(1,5) if x+y <6]`  
[2, 5, 10, 17, 3, 6, 11, 4, 7, 5]
- ii. Write a Python program to produce the same result without using list comprehension. (5 marks)

```
L = [ ]
for x in range(1,5):
    for y in range(1, 5)
        if x+y < 6:
            L.append(x+y**2)
```

4. What are printed in the following programs: (6 marks each)

i. `def double(a):`  
`a = a * 2`  
`print(a)` [10, 10]

`b = [10]`  
`double(b)`  
`print(b)` [10]

ii. `def double(a):`  
`a[0] = a[0] * 2`  
`print(a)` [20]

`b = [10]`  
`double(b)`  
`print(b)` [20]

5. Given the definition of the list L

`L = [1, 2, 3, (1, 2, 3), "123", [1, 2, 3]]`

State the values of the variables on the left of the statements below. Where there is an error, state why the error occurs. You do not need to provide corrections. (2 marks each)

i. `A = L[-1:-4:-2]` [[1, 2, 3], (1, 2, 3)]

ii. `B = L[2] + L[3][2]` 6

iii. `C = L[2] + L[4][2]` Error, can't add integer and string.

iv. `D = L[3] + L[5]` Error. Can't add tuple and list.

v. `E = L[1] * L[4][1]` "22"

vi. `F = len(L[2]) + len(L[4])` Error. len() cannot operate on an integer (L[2]).

6. What is the value of `L` upon the execution of the following program if the input string is "Year 2022"? (10 marks)

```
s = input("Enter a string: ")
L = list(s)
for i in range(0, len(L)//2):
    L[i], L[-i-1] = L[-i-1], L[i]
```

["2", "2", "0", "2", " ", "r", "a", "e", "Y"]

7. Given a 2D vector  $(x, y)$ , where  $x$  and  $y$  are floating point numbers, the magnitude of the vector is  $\sqrt{x^2 + y^2}$  and the angle of the vector is given by  $\text{atan}(y/x)$ . If  $x$  is very small, the angle is  $\pi/2$ . The following function takes a tuple of two elements, `V`, representing  $(x, y)$ , and a variable, `eps`, defaulted to  $10^{-8}$ , as the inputs and returns two values: the magnitude and the angle. You may assume that the statement `import math` is in place before the function.

```
def vec_mag_angle(V, eps = 1e-8):
    magnitude = math.sqrt(V[1]*V[1] + V[2]*V[2])
    return magnitude
    if V[1] < eps: # calculate angle, but check x value first
        angle = atan(V[1]/V[2])
    else:
        angle = pi/2
    return angle
```

- i. But the function contains errors. Circle the errors and provide the full correct program below. You may omit the comments in your program. (10 marks)

```
def vec_mag_angle(V, eps = 1e-8):
    magnitude = math.sqrt(V[0]*V[0] + V[1]*V[1])
    if abs(V[0]) > eps: # calculate angle, but check x value first
        angle = math.atan(V[0]/V[1])
    else:
        angle = math.pi/2
    return magnitude, angle
```

- ii. Write a statement to call the function with the returned value properly assigned. You may supply your own parameter values; if you use variables, they must be defined. (3 marks)

`m, a = vec_mag_angle((3, 5))`

This solution is not unique. Student can supply any values for the vector, and need to supply a very small value for `eps` if not omitted. The left hand side can have one or two variables.

8. Given the function

```
def get_item (data, item = -1):
    return data[item]
```

What do the following lines of code print? Where there is an error, state what the error is, provide the correction and state what is printed based on your correction. (3 marks each)

```
aList = [1, -2.0, (3, 4), True, "True"]

print(get_item([aList], 0)) # [1, -2.0, (3, 4), True, "True"]

print(get_item(aList[-1])) # "e"

print(get_item("aList", 5)) # Error. Index out of range. Change 5 to 4. Prints "t".
                             # Other corrections possible.
```

9. i. Construct a function called `factors(i)` which returns the factors of the integer parameter `i` in a list in ascending order. For example, if `i` is 16, the returned list is `[1, 2, 4, 8, 16]`. (6 marks)

```
def factors(i):  
    L = []  
    for num in range(1, i+1):  
        if i%num == 0:  
            L.append(num)    # or L = L + [num]  
    return L
```

Or

```
def factors(i):  
    return [num for num in range(1, i+1) if i%num == 0]
```

- ii. The function below, `hcf(i1, i2)`, returns the highest common factor of its two integer parameters `i1` and `i2` using `factors()`. For example, if `i1 = 20`, and `i2 = 30`, `hcf()` returns 10. If `i1 = 36`, and `i2 = 18`, `hcf()` returns 18. If `i1 = 20`, and `i2 = 13`, `hcf()` returns 1. Fill in the blanks to complete the function. (6 marks)

```
def hcf(i1, i2):  
    L1 = factors(i1)    # L1 contains factors of i1  
    L2 = factors(i2)    # L2 contains factors of i2  
  
    for item in L1[::-1]:    # check from the right  
  
        if item in L2:  
  
            return item
```

Note: L1 and L2 in the for loop and the if statement can swap places.

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