Your Name: _____ Group number: _____

MA1008 Introduction to Computational Thinking
Quiz 2, Week 9, Semester 1. Answer all the questions in the spaces provided.
Time allowed: 1 hour

---

1. i. Write one Python list comprehension statement that produces the following list:
   `[1 4 16 25 49 64 100]` (5 marks)
   ii. Write another Python list comprehension statement that produces the following list:
   `[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]` (5 marks)

2. When given these two assignment statements
```
L = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]
L[1][1] = -2
```
   i. The second assignment statement fails? Why? (5 marks)
   ii. Rewrite the first statement with minimal changes such that the second can be performed.
```
L = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]
```
(5 marks)

3. i. Without using `int( )`, write a function called `digitvalue(S)` where `S` is a string containing one digit character only as the input parameter and returns the integer value of the character. You can assume that the given string contains only one digit. (5 marks)

   ii. Without using the string methods `isupper()`, `islower()` or `isalpha()`, write a function called `letters(S)` which returns True if `S`, a string, contains only letters of either case, and False otherwise. (5 marks)

4. The program below defines a function that takes in an object, modifies its contents, and returns the object. The function is then called in a statement that passes in a list as the input object.

```
def modify_object(object, item = 0):
   object = object.append(item)
   object[0] = object[1]
   return object

new_object = modify_object([1, 2, 3])
```

   i. There is a problem in this program. What is it? (5 marks)
   ii. Correct the problem by modifying the call statement or the function. (5 marks)

5. Given a 2D vector (x, y), the magnitude of the vector is $\sqrt{(x^2 + y^2)}$ and the angle of the vector is given by atan(y/x). The following function is defined to take a vector as the input and return two values: the magnitude and the angle.

```
def magnitude_angle(x, y, eps = 1e-10):
   magnitude = math.sqrt(x*x + y*y)   # calculate magnitude
   return magnitude
   if abs(x) < eps:   # calculate angle, but check x not 0
      angle = atan(y/x)
   else:
      angle = pi/2
   return angle
```

i. But the function contains errors. Correct the errors by providing the full correct program. You may omit the comments in your program. To ease your task, you may copy and paste the program into the solution area and edit out the errors. (7 marks)

ii. Write a statement to call the function. You may supply your own parameter values, taking care of the fact that, if you use variables, they must be defined. (3 marks)

6. Given the tuple `T = (1, 2, 3, 4)`,

i. write a statement that uses `T` to form a new tuple `T2 = (1, 2, 3, 4, 5)` (4 marks)

ii. write another statement that uses `T` to form another new tuple `T3 = (1, 2, 3, [4, 5])` (6 marks)

7. Fill in the blanks of this program

```
interest = "Interest"
saving = 6017.29
interest_rate = 0.015
print("{____}=${____}".format(saving*interest_rate, interest)
```

such that the statement printed, using □ to denote the space character, is

```
□□Interest=$90.26□□
```

(10 marks)

8. A dictionary is set up in the following statement to store the names of the weekdays using only the first letter of the name as the key, but it has errors.

```
weekday = {M:"Mon", T:"Tue", W:"Wed", T:"Thu", F:"Fri", S:"Sat", S:"Sun"}
```

i. Identify and explain the errors, and provide the correct statement. (5 marks)

ii. Using your corrected dictionary, write a for loop to print the weekdays as follows: (5 marks)

```
Mon Tue Wed Thu Fri Sat Sun
```

9. i. Construct a dictionary that stores the names of the months in a year using an appropriate key.

ii. Construct another dictionary that stores the number of days in a month using the same key. (5 marks for i and ii)

iii. Using these two dictionaries, write a for loop that prints the name of a month against the number of days in the month for all the 12 months, as follows: (5 marks)

```
January 31
February 28
March 31
   . . .
   . . .
December 31
```

10. The following function `substring()` has four parameters, `S`, `sub`, `start` and `end`. It checks if `sub`, a string, is contained between the indices `start` and `end`, inclusive, of another string `S`, with `start` defaulted to 0, and `end` defaulted to `-1`, which means the full length of S. If `sub` is contained in `S`, the function returns the index of the character in `S` where `sub` starts. It returns `-1` otherwise. For example, if `S = "A nice idea"` and `sub = "nice"`, the statement

```
index = substring(S, sub)
```

would assign `2` to index. If `sub = "abc"`, then `index` would get the value `-1`. It uses an internal function `is_sub()` to accomplish the task. Complete the program by filling in the blanks.

(10 marks)

```
def substring(S, sub, start=0, end=-1):

    def is_sub(S, sub, start, end):
        # Function checking if sub exists in S at start
        # Returns True if it does, False otherwise
        if start + len(sub) > end+1: # check S has enough characters
            return _____________
        for i in range(len(sub)):   # Go through the length of sub
            if S[i+start] != sub[i]: # Check character by character
                return _____________
        return _____________

    if end == -1 or end > len(S):  # end = -1 means full length
        end = len(S)               # Set end to full length of S
    for i in range(start, end+1):  # Move up S one character at a time
        if is_sub(S, sub, i, end):  # Check if sub starts at i
            return __________  # substring found
    return __________          # substring not found
```