MA1008 Introduction to Computational Thinking Quiz 1
Answer all the nine questions in the spaces provided
AY 2022/2023, Semester 2, Week 5

Your Name: ________________________________ Group: _____

Solutions

The answers given here may not be unique. If a student's answer is different, do check whether it is a correct alternative.

1. The variable `num` carries a positive integer value. Write a for loop that prints all the factors of `num` in one line separated by a comma and space, except the last factor, which should not have a comma after it. For example, if `num = 24`, your statements should print

```
    1, 2, 3, 4, 6, 8, 12, 24
```

(10 marks)

```
for i in range(1, num):   # check all numbers except num itself
   if num%i == 0:
      print(i, end = ", ")
print(num)   # print last number without trailing comma
```

Alternatively

```
for i in range(1, num+1):   # go through all numbers
   if num%i == 0:
      if i == num:   # check if it's the last number
         print(num)
      else:
         print(i, end = ", ")
```

2. Given `A = 5, B = -4, C = 2, D = 3.0, E = 10, P = True, Q = True, R = False`, write down (a) the value and (b) the data type of each of the following expressions: (10 marks)

   i. `A + B * - C ** D ** C // E` 209.0, float

   ii. `A * B > C + D and P and not Q or R and not A` False, Boolean

3. Fill in the blanks in the following program such that it prints a triangular pattern of * and its reflection given a value of `n`. Below are two examples: (12 marks)

```
n=3   *****                          n=4  *******
       ***                                 *****
        *                                   ***
       ***                                   *
      *****                                 ***
                                           *****
                                          *******
```

```
n = int(input('Enter n: '))
for i in range(n, 0, -1):
   print(' ' * (n-i), '*' * (i*2-1))
for i in range(2, n+1):
   print(' ' * (n-i), '*' * (i*2-1))  # Other solutions possible.
```

4. One could represent a date in Python using three integers for the year, month and day. How would you represent the following items in Python? (3 marks each)

   i. The time of the day in the 12-hour format, with hour, minute and am or pm. 2 integers, one string

   ii. The time recorded for completing a marathon, in hours, minutes and seconds up to two decimal places. 2 integers, 1 float

   iii. The address of a house, which consists of the house number, the street name, and the post code preceded by the word Singapore.
   1 integer, 2 strings and 1 integer Or
   3 strings and 1 integer Or 4 strings
   Note: House number could be string, like 2a

5. There are three types of residents in Singapore: citizen, permanent resident and foreigner, designated by the strings "C", "PR" and "F" respectively. The residents are categorised into three age groups: child, adult or senior, designated by the strings "C", "A", and "S" respectively. A hospital provides discounts to residents for medical treatment using the following rules:
   a. Citizen: Children and seniors, 75% discount. Adults: 50% discount.
   b. Permanent resident: Children and seniors, 50% discount. Adults: 25% discount.
   c. Foreigner: no discount for all age groups.

   The variables Typ and AG carry the type and age group of a resident respectively and always carry valid values as given above. The program below prints the discount given. But it contains errors and redundancies. Rewrite the program correctly and without the redundancies. (12 marks)

```
if Typ == "C" and AG == "C" or "S":
   print("Discount =", 75%)
elif Typ == "C" and AG == "A":
   print("Discount =", 25%)
elif Typ == "PR" and AG == "C" or "S":
   print("Discount =", 50%)
elif Typ == "PR" and AG == "A":
   print("Discount =", 25%)
elif Typ == "F" and AG == "C" or "A" or "S":
   print("No discount")
```

```
if Typ == "C" and (AG == "C" or AG == "S"):
   print("Discount = 75%")
elif Typ == "C" and AG == "A":
   print("Discount = 25%")
elif Typ == "PR" and (AG == "C" or AG == "S"):
   print("Discount = 50%")
elif Typ == "PR" and AG == "A":
   print("Discount = 25%")
else:  # only foreigner left, conditions here are redundant
   print("No discount")
```

There are three distinct types of errors here (the missing brackets, missing AG == and the output strings in the print statement), each occurring more than once, plus the redundancy in the last elif, which should just be else. So, four types of errors in all, each 3 marks. If a student doesn't identify all occurrences of a type, then deduct part of the 3 marks.

6. In the imperial system for weight, 1 stone = 14 pounds and 1 pound = 16 ounces.

   i. The variable `stones` carries a floating point value which is the weight in stones. Write Python statements to convert this value to its equivalent in terms of stones, pounds and ounces. For example, 35.229 stones is 35 stones, 3 pounds and 3.296 ounces. (You do not need to include `input()` or `print()` statements.) (8 marks)

```
stones = 5.29   # actual value is arbitrary here, not needed
int_stones = int(stones)
pounds = (stones - int_stones)*14
int_pounds = int(pounds)
ounces = (pounds - int_pounds)*16
```

The above is adequate, as the answer is `int_stones`, `int_pounts`, `ounces`. Instead of using int(pounds), some students may us pounds//1 to get the integer value. This is not good enough because pounds//1 delivers the integer value as a float. Take off two marks for that. No need for a print statement to print the result.

   ii. The three variables `stones`, `pounds` and `ounces` store the weight of an object in terms of the three units (such as 2 stones 5 pounds and 9.43 ounces). Write Python statements to convert this weight to its equivalent in pounds only. (4 marks)

```
equi_pounds = stones*14 + pounds + ounces/16
```

7. Given the following program:

```
counter = 0
value = 0
while True:
    counter += 1
    if counter >= 100:
        break
    if counter % 7 > 0:
        continue
    if counter % 7 == 0:
        value += counter
```

   i. State what the program does. (3 marks)

This program sums all the positive numbers less than 100 and divisible by 7.

   ii. Rewrite the `while` loop to achieve the same result without using `break` or `continue`. (8 marks)

```
while counter < 100:
    counter += 1
    if counter % 7 == 0:
        value += counter
```

This may be done using a for loop also.

```
for counter in range(1, 100):
    if counter % 7 == 0:
        value += counter
```

8. In 4D lottery, an integer n is printed with four digits. If n has less than four digits, then the leading positions are printed with 0. For example, 12 is printed as 0012 and 123 is printed as 0123, while 1234 is printed exactly as it is. Write Python statements to perform the printing given n, where 0 < n <= 9999. (You may consider using the + operator for joining two strings, for example, "AB" + "C" = "ABC".) (12 marks)

```
if n < 10:
   print("000" + str(n))
elif n < 100:
   print("00" + str(n))
elif n < 1000:
   print("0" + str(n))
else:
   print(n)
```

9. Write Python statements for the following mathematical equations. You may assume that all the variables carry appropriate values and all the required constants and functions have been imported from their libraries, hence you do not need to import the library or provide the library prefix to the constants and functions. (6 marks each)

i. $$S = A(\frac{\sin (\alpha\sqrt{(x^2 + y^2)h^3}}{\cos\left(\frac{y}{x}+2h^2\right)})$$

```
S = A*sin(a*sqrt((x*x+y*y)*h**3))/cos(y/x+2*h*h)
```

ii. $$t = a_0 + \sum_{i=1}^{n}(a_1 \cos\frac{i\pi x}{L}(\sum_{j=1}^{i}\frac{1}{j}))$$

```
t = a0
for i in range(1, n+1):
   s = 0
   for j in range(1, i+1):
      s += 1/j
   t += a1*cos(i*pi*x/L)*s
```