UECM1703 Test Marking Guide

Name: Student ID: Mark: /**17**

Course Code & Course Title:	UECM1703 Introduction to Scientific Computing		
Faculty:	LKC FES, UTAR	Course:	AM, FM
Session:	Oct 2022	Lecturer:	Liew How Hui

Instruction: Answer all questions in the space provided. **If you do not write your answer in the space provided, you will get ZERO mark**. An answer without working steps may also receive ZERO mark.

CO3: Write program scripts for mathematical software

1. (a) Write down three basic data types/structures which we often use when writing scripts in Core Python for mathematical software. In no less than 20 words, comment on one of the most important basic data type in relation to real-world applications. (2 marks)

   Ans. Any 3 from string, Boolean, integer, floating point number, tuple, list or function would be OK. . . . . [1.5 marks]

   The basic data type list is the most important because it allows us to store data of any data type by appending and we can retrieve the data by indexing. . . . . [0.5 mark]

   (b) Write a Python program script to implement the following function

$$f_1(x) = 2\Big(\frac{\pi^2}{3} - 4\cos x\Big) + 5\Big(2\sin x\Big), \quad -\pi < x < \pi$$

   by

   (i) **defining the function** $f_1(x)$ in Core Python with appropriate imports from the Core Python mathematical module. Note that x is in radians and no unit conversion is necessary; (1 mark)

   (ii) **writing the Python command** which prints $f_1(3)$ to 6 decimal places and **writing down the value of** $f_1(3)$ **that is calculated using your scientific calculator** for the verification of the correct implementation of $f_1(x)$. (1 mark)

   Ans. A sample Python scripts implementing item (i) and (ii) is

```
# Part (i)
from math import sin, cos, pi                      # [0.5 mark]
def f_1(x):
    return 2*(pi**2/3 - 4*cos(x)) + 5*2*sin(x)     # [0.5 mark]

# Part (ii)
print("f_1(3)={:.6f}".format(f_1(3)))              # [0.5 mark]
# Calculator result: 15.910876                     # [0.5 mark]
```

(c) Write a Python program script (using for loop or otherwise), to implement the **definition of a Python function** for the function

$$f_{20}(x) = 2\Big(\frac{\pi^2}{3} + \sum_{n=1}^{20} \frac{4(-1)^n}{n^2} \cos nx\Big) + 5\sum_{n=1}^{20} \frac{2(-1)^{n+1}}{n} \sin nx$$

for $-\pi < x < \pi$. (2 marks)

Ans. A sample Python scripts implementing item (i) and (ii) is

```
from math import sin, cos, pi
def f_20(x):
    first_term = pi**2/3
    for n in range(1,20+1):
        first_term = first_term + 4*(-1)**n/n**2*cos(n*x)
                                                # [0.8 mark]
    second_term = 0.0
    for n in range(1,20+1):
        second_term = second_term + 2*(-1)**(n+1)/n*sin(n*x)
                                                # [0.7 mark]
    return 2*first_term + 5*second_term         # [0.5 mark]
```

(d) Write a Python script by using the while loop to find the largest value of the following series not exceeding 110:

$$\frac{5}{7} + \frac{7}{10} + \frac{9}{13} + \frac{11}{16} + \frac{13}{19} + \ldots$$

where the denominator sequence and the numerator sequence are arithmetic progression sequence. The Python script must give the total sum and the number of terms in the series. (3 marks)

Ans. An implementation using while loop is shown below.

```
n = 0
thesum = 0
numer  = 5
denom  = 7                             # [0.5 mark]
while thesum < 110:                    # [0.5 mark]
    lasterm = numer/denom
    thesum += lasterm
    numer  += 2
    denom  += 3
    n      += 1                        # [1.5 marks]
print("The sum is", thesum - lasterm)
print("Number of terms, n =", n-1)     # [0.5 mark]
```

CO1: Perform vector and matrix operation using computer software

2. (a) Write down the necessary imports and the single Python command to construct the matrix

$$\begin{bmatrix} 1-2^6 & (\sqrt{7}+1)^5 \\ \cos\frac{\pi}{2} & e^{2\pi} \end{bmatrix}$$

as an array of floats in the variable `A`. (1.5 marks)

Ans. Marks will be deducted for wrong expressions.

```
from math import cos, pi, exp, sqrt               # 0.3 mark
A = np.array([                                    # 0.3 mark
  [1-2**6, (sqrt(7)+1)**5],                       # 0.5 mark
  [cos(pi/2), exp(2*pi)]])                        # 0.4 mark
```

(b) Given the matrix of random integers below:

$$C = \begin{bmatrix} 92 & 73 & 98 & 78 & 18 & 34 \\ 27 & 11 & 60 & 61 & 28 & 60 \\ 25 & 25 & 24 & 69 & 55 & 37 \\ 73 & 77 & 23 & 97 & 20 & 27 \\ 46 & 64 & 87 & 84 & 82 & 38 \end{bmatrix}$$

(i) Write down the output of `print( C[:4, [0, 2, 1, 3]] )` (1.5 marks)

Ans.

$$\begin{bmatrix} 92 & 98 & 73 & 78 \\ 27 & 60 & 11 & 61 \\ 25 & 24 & 25 & 69 \\ 73 & 23 & 77 & 97 \end{bmatrix}$$

[1.5 marks]

(ii) Write down the output of `print(C[::-2,::-1])` (1 mark)

Ans.

$$\begin{bmatrix} 38 & 82 & 84 & 87 & 64 & 46 \\ 37 & 55 & 69 & 24 & 25 & 25 \\ 34 & 18 & 78 & 98 & 73 & 92 \end{bmatrix}$$

[1 mark]

(iii) Write down a single-line command to obtain the array view below from the array C.

$$\begin{bmatrix} 27 & 11 & 60 \\ 25 & 25 & 24 \\ 73 & 77 & 23 \end{bmatrix}$$

(1 mark)

Ans. C[1:4, 0:3] .... [1 mark]

(iv) Write down the Python command(s) using a for loop or otherwise to change the diagonal of C to 120 as follows:

$$\begin{bmatrix} 120 & 73 & 98 & 78 & 18 & 34 \\ 27 & 120 & 60 & 61 & 28 & 60 \\ 25 & 25 & 120 & 69 & 55 & 37 \\ 73 & 77 & 23 & 120 & 20 & 27 \\ 46 & 64 & 87 & 84 & 120 & 38 \end{bmatrix}$$

(1 mark)

Ans. for i in range(C.shape[0]): C[i,i]=120 ....[1 mark]

(c) Write a Python script to first generate the following sequence

9, 19, 29, 39, 49, 59, 69, 79, 89, 99, 101, 201, 301, 401, 501, 601, 701, 801, 901

and store it in the variable `x`. Then a matrix A is generated as a Numpy array with the first column being the values from the variable `x`, the second column being the application of the function

$$f(x) = \ln \frac{x^2}{x^2+1}$$

on the variable `x` elementwise, the third column being the value $g(x)$ of the approximation of $f(x)$ elementwise on `x`:

$$g(x) = -\frac{1}{x^2+1} - \frac{1}{2}(\frac{1}{x^2+1})^2 - \frac{1}{3}(\frac{1}{x^2+1})^3$$

and the fourth column is the absolute difference between the second column and the third column. The printout the matrix A in Python should look like

```
[[ 9.00000000e+00 -1.22700926e-02 -1.22700870e-02  5.58398414e-09]
 [ 1.90000000e+01 -2.76625349e-03 -2.76625348e-03  1.45904300e-11]
 [ 2.90000000e+01 -1.18835427e-03 -1.18835427e-03  4.97860867e-13]
 [ 3.90000000e+01 -6.57246162e-04 -6.57246162e-04  4.65909863e-14]
 [ 4.90000000e+01 -4.16406419e-04 -4.16406419e-04  7.52354993e-15]
 [ 5.90000000e+01 -2.87232517e-04 -2.87232517e-04  1.65769091e-15]
 [ 6.90000000e+01 -2.10017852e-04 -2.10017852e-04  4.49808376e-16]
 [ 7.90000000e+01 -1.60217897e-04 -1.60217897e-04  1.24818775e-16]
 [ 8.90000000e+01 -1.26238718e-04 -1.26238718e-04  7.25602304e-17]
 [ 9.90000000e+01 -1.02025200e-04 -1.02025200e-04  1.61139548e-17]
 [ 1.01000000e+02 -9.80248004e-05 -9.80248004e-05  8.74138002e-18]
 [ 2.01000000e+02 -2.47515563e-05 -2.47515563e-05  4.90161026e-17]
 [ 3.01000000e+02 -1.10373449e-05 -1.10373449e-05  3.00696696e-17]
 [ 4.01000000e+02 -6.21884746e-06 -6.21884746e-06  1.22226854e-17]
 [ 5.01000000e+02 -3.98403994e-06 -3.98403994e-06  3.59396080e-18]
 [ 6.01000000e+02 -2.76853778e-06 -2.76853778e-06  4.94243725e-18]
 [ 7.01000000e+02 -2.03499582e-06 -2.03499582e-06  5.06606171e-17]
 [ 8.01000000e+02 -1.55859985e-06 -1.55859985e-06  2.53148502e-17]
 [ 9.01000000e+02 -1.23182822e-06 -1.23182822e-06  4.65349313e-17]]
```

(2 marks)

Ans. A sample Python script is shown below:

```
seq1 = np.arange(9,109,10)
seq2 = np.arange(101,1001,100)
x = np.hstack((seq1, seq2))                        # [0.5 mark]
col2 = np.log(x**2/(x**2+1))                       # [0.5 mark]
col3 = -1/(x**2+1) - (1/(x**2+1))**2/2 - (1/(x**2+1))**3/3
col4 = np.abs(col2-col3)                           # [0.5 mark]
A = np.vstack((x,col2,col3,col4)).T                # [0.5 mark]
```