LAB TEST 1: CITS2401 Computer Analysis and Visualisation
Semester 2, 2017

SESSION: Tuesday

Exam time 120 minutes
Maximum Marks: 40

- Please follow all instructions carefully.
- Submit your answers on LMS during the test session.
- You must include all files in a single .zip file.
- Please ensure that you submit the correct files and complete submission process.
- If you do not follow the submission guidelines, your submission will be graded zero.

Mobile phones are NOT allowed.
Two-way internet communication is NOT allowed.

Submission

- Include your solution files in one folder. Right click on the folder and hover/click on "*Send To"* option, then click on "*Compressed (zipped) folder*". This will create a zip file.

- Rename you zip file to YourSurname_StudentNo.zip

- Submit the zip file via the CITS2401 page on LMS http://www.lms.uwa.edu.au. Weekly Modules → Week 8 → Lab Test 1

- Validate your submission by downloading it again and checking the files.

LAB TEST 1: CITS2401 Computer Analysis and Visualisation
Semester 2, 2017

Question 1: (10 marks)

Consider a square (4 x 4) matrix A such as

$$A = \begin{bmatrix} 12 & 3 & 9 & 13 \\ 5 & 6 & -12 & 12 \\ 7 & 10 & 32 & 0 \\ -9 & 15 & 1 & 19 \end{bmatrix}$$

Create a script file and write command(s) for the following tasks in the script file. The script should work for any square matrix A of size [4 x 4], not just the example above.

All tasks are independent of each other. Don't forget to mention the task as the comment in the script file before its solution.

1. Swap the first and last row of matrix A.

2. Add the first and third column of matrix A and store the resultant column as fifth column in matrix A.

3. Sum all the elements in the matrix A.

4. Replace the element at position (2nd row and 4th column) of matrix A by element at position (4th row and 1st column) of matrix A.

5. Extract the diagonal elements of matrix A and store them as a vector B. [Hint: Diagonal elements are at position (1,1), (2,2),… and so on.]

SAVE

- Save your script file with name YourSurname_studentnumber e.g. `Potter_456789876`

LAB TEST 1: CITS2401 Computer Analysis and Visualisation
Semester 2, 2017

Question 2: (20 marks)

In this problem you will be modelling a Lotka–Volterra Predator-Prey model. It is comprised of a pair of first-order, non-linear differential equations frequently used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as a prey.

Consider the following difference equation which explains the relationship between the population of sharks and other fish in the ocean.

$$S_{n+1} = S_n + (a\,S_n - b\,S_n F_n)/100 \qquad (1)$$

$$F_{n+1} = F_n + (c\,S_n F_n - d\,F_n)/100 \qquad (2)$$

where *S* and *F* represent the population of shark and fish respectively, *n* represents the generation number while *a*, *b*, *c* and *d* are the constants having positive values.

The inputs to the function are: initial population of shark S_0 , initial population of fish F_0 , number of generations `g` and values for constants `a`, `b`, `c` and `d`. The required output of the function are: vectors `S` and `F` containing populations of shark and fish at each generation respectively, and the maximum populations `Smax` and `Fmax` of both shark and fish respectively over `g` generations. Remember the first generation is considered as initial population provided as S_0 and F_0 and must be included in the output vectors `S` and `F` respectively.

The first line of the function must be

```
function [Smax,Fmax,S,F] = sharkfish(S0,F0,g,a,b,c,d)
```

Include, as a comment in the second row of your function file: your full name and student number.

The sample input and output to test your code is provided below (ignore the format and concentrate on matching the result values)

```
>> [Smax,Fmax,S,F] = sharkfish(55,75,900,2,0.1,0.2,5)
Smax = 111.4921
Fmax = 167.7840
S = Array of [1x900]
F = Array of [1x900]
```

SAVE

- Save your function with name `sharkfish.m`

LAB TEST 1: CITS2401 Computer Analysis and Visualisation
Semester 2, 2017

Question 3: (10 marks)

Consider a structure array `student` having fields: `first_name`, `last_name`, `student_ID` and `marks`, which contains record of `m` number of students enrolled in a course. The field `marks` is a matrix of size [1 x n] containing the marks of `n` different quizzes of the course.

Write a function called `average` which takes the structure `student` as an input and returns the structure with same data as `student` with an extra added field `ave` in the structure, which contains the calculated average of marks of different quizzes obtained by each student in the course.

The first line of the function must be

```
function student2 = average(student)
```

Include, as a comment in the second row of your function file, your full name and student number.

The sample input and output to test your code is provided below (ignore the format and concentrate on matching the result values), however your code should work with any general input data having correct format.

```
>> struct2table(student)

first_name    last_name    student_ID       marks
__________    _________    __________    _____________

'Allan'       'Norris'     21111111      [10,20,30]
'Alex'        'Collie'     22222222      [5,10,25,60]

>> student2 = average(student);
>> struct2table(student2)

first_name    last_name    student_ID      marks          ave
__________    _________    __________      _______        ____

'Allan'       'Norris'     21111111      [10,20,30]       20
'Alex'        'Collie'     22222222      [5,10,25,60]     25
```

SAVE

- Save your function with name `average.m`