UNIVERSITY OF SURREY

ENG0018 Computer Laboratory Module **2023**

Matlab coursework tasks: overall 30% weighting

Submission deadline

This coursework assessment comprises 3 tasks. Complete and upload your coursework submission as a zipped folder (2 × matlab script **.m** files – using the supplied template files plus 1 × Word document containing your answers to questions) on SurreyLearn ***before*** **16:00 on Wednesday 22nd November 2023**.

Your submitted work and answers must be your own individual work

You will have approximately three weeks to complete this coursework in your own time. You are strongly advised to start the tasks early so that you can resolve any problems before the submission deadline. It is far more effective to spend a bit of time each day working on the tasks than trying to do it all at the last minute.

There is no limit to the amount of time you may spend on the coursework but if you have attended the labs and completed worksheets 01-05 it should not take more than about 8 hours to complete all of the tasks to a good standard. The tasks are based heavily on the content of lab worksheets 4 (tasks 1 and 2) and 5 (task 3) and it is essential to complete these before attempting the coursework. The coursework questions are worth a maximum of 8% and your .m file submissions with working code are worth up to a maximum of 22%, giving a total of 30%.

The code in your submitted .m files must be;

(1) executable - it must run and produce output, even if that output is incorrect and

(2) comprehensible - it ***must contain your comments*** indicating that you understood what your code did and the comments convey this in such a way that it is understandable to the examiners.

(3) in the provided files – use the supplied files inside the Matlab coursework folder on SurreyLearn; **T1T20000000.m** and **T30000000.m** replace the characters "0000000" with your own personal URN (ID number). Save the file with your URN and complete your coursework tasks in the appropriate file as per the instructions. There is an example in the **T1T20000000.m** file with some information that will help you to complete tasks 1 and 2, including guidance on commenting. There are also coursework questions connected with the example.

Marks are allocated for commenting and if there are no comments or your code does not run you will score 0% for that component. However, you may still be able to score marks on the coursework questions.

Assessed learning objectives

LO	Description	Task
2.1	Create, save and edit a script .m file	1, 2, 3
2.2	Add comments to a script	1, 2, 3
3.2	Perform simple operations on matrices using conditional statements and loops	1, 2
4.1	Create a binary image from a 2-dimensional binary [0 1] matrix	1
4.2	Create a colour (RGB) image with a colour gradient, from a 3-dimensional array	2
5.2	Extend a 1-dimensional random walk model to include boundary conditions and bias	3

Matlab coursework tasks: overall 30% weighting

Task summary

The tasks have been designed to assess the problem solving skills and programming skills that you have developed on the Matlab component of the Computer Laboratory Module. The tasks have a strong emphasis on spatial thinking as well as logical and Mathematical thinking. In assessed task 1, you will create a binary image in a loop, according to a specified set of conditions. In task 2, a colour image with colour gradients will be created. In task 3 you will model an online gambling process and estimate how much money you might make from the players.

Preliminary work

Download the provided file for tasks 1 and 2 - T1T20000000.m from Surreylearn and save the file in your **One Drive** area with the characters "0000000" replaced with your **URN** e.g. **URN1234567.m** Use this file to complete coursework tasks 1 and 2. Do the same for the file for task 3 using the file T30000000.m.

Read the additional information contained in the provided files along with the task steps in this document and carefully follow any instructions given in the file and in this coursework tasks document.

Download the Matlab coursework questions Word document and save the file in your One Drive area with a name that includes your **URN** at the start of the filename, e.g. **1234567_Matlab.docx**

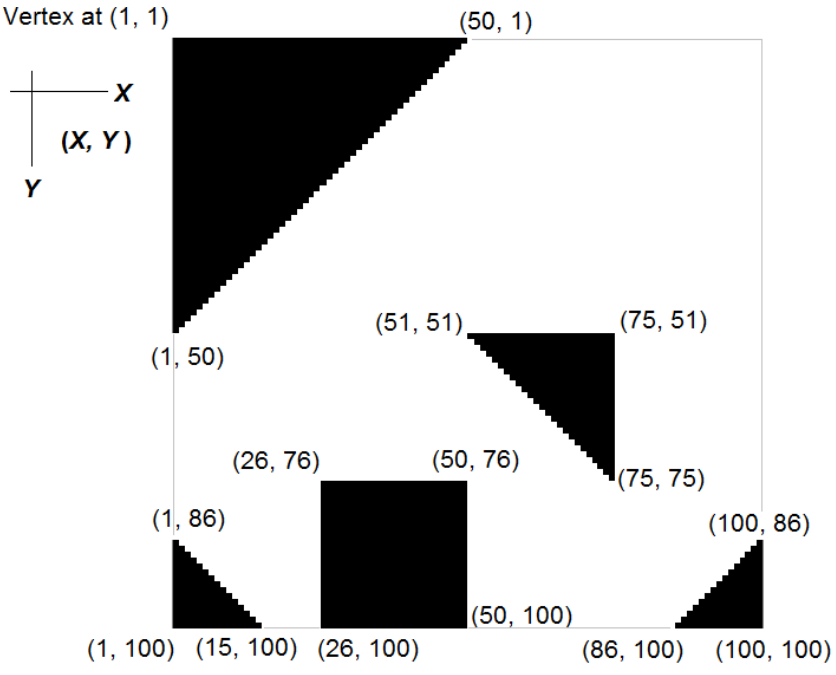
Do **NOT** put your name anywhere in your coursework submission

You must upload **both** of the Matlab script m-files and the completed coursework question sheet ***before*** **16:00 on Wednesday 22nd November 2023.**

Important

🖫 Make sure you save you work regularly on OneDrive and keep a back-up on a portable storage device.

Coursework Task 1

Step	Task 1 Instructions (* indicates your work will not be marked if this is not included) **Binary image creation and basic operations in Matlab**	Mark(s) %
1	Rename and save the provided T1T20000000.m file as instructed in the preliminary work section. Read the information in the file then run the example code for task 1, step 1. Answer **questions 1-7** on the question sheet. You can answer these questions at any time.	**5**
2	Create a 2-dimensional array of 'ones' (size 100×100) in the indicated area for task 1 in your m-file. Using the information provided in figure 1 below set up a **single** '**for**' loop with 100 iterations from row 1 to row 100 and apply appropriate conditional statements **inside of the loop**, to generate the binary image in figure 1. *When your code runs, your output must include a figure which shows how the image builds up row-by-row (as in the m-file example code). The changing row number must also be displayed to the left side of the image, as in the m-file example. Set the pause time as **pause(0.01)**. Coordinates in figure 1 represent the single black pixel at the vertex of the shape. The (X, Y) image coordinate frame is defined as; X starts at 1 (left most column of pixels) and increases to the right. Y starts at 1 (topmost row of pixels) and increases downward. The image coordinates do **not** correspond to the image array (matrix) indices. For example, in an array A, the pixel at the (X, Y) vertex $(75, 51)$ will have array indices $A(51, 75)$ where 51 is the row index and 75 the column index.  **Figure 1**	6 1

3	***Add a few short comments accurately stating the important stages of your process. E.g., state the purpose of the conditional tests and what is happening in the code inside each conditional test.	1
4	Write additional code to calculate the total number of black pixels in your image and to display the total number of black pixels beneath the image in the figure window.	1
5	Write additional code to compute the total number of black pixels below (and including any black pixels on) the image diagonal from (1, 1) to (100, 100). Write code to display your answer in the figure window just below where you have displayed the answer in step 4. Now answer Matlab coursework task questions **8 and 9**	1 **1**
	Total marks =	**16 %**

Now start task 2

Coursework Task 2

Step	Task 2 Instructions (* indicates your work will not be marked if this is not included) **Colour image creation and basic operations in Matlab**	Mark(s) %
1	Deactivate your entire block of code for task 1 using %{ and %}	
	Create a 3-dimensional uint8 array of zeros (size $100 \times 100 \times 3$) in the indicated area for task 2 in your m-file. Using the information in figure 2 and the summary of colour gradients below, set up a **single** '**for**' loop with 100 iterations from row 1 to row 100 and apply appropriate conditional statements **inside of the loop**, to generate the 100×100 pixel colour image in figure 2. Note, the small yellow and pink boxes in figure 2 are not part of the image, they are to clarify the pixel indices and colour values in the image regions where the vertices of shapes of different colours meet. All pixels on the diagonal are green. Image (X, Y) coordinates of vertices are shown, as in task 1.	5
	*When your code runs, your output must include a figure which shows how the image builds up row-by-row (as in the m-file example code). The changing row number must also be displayed to the left side of the image, as in the m-file example. Set the pause time as **pause(0.01)**.	1

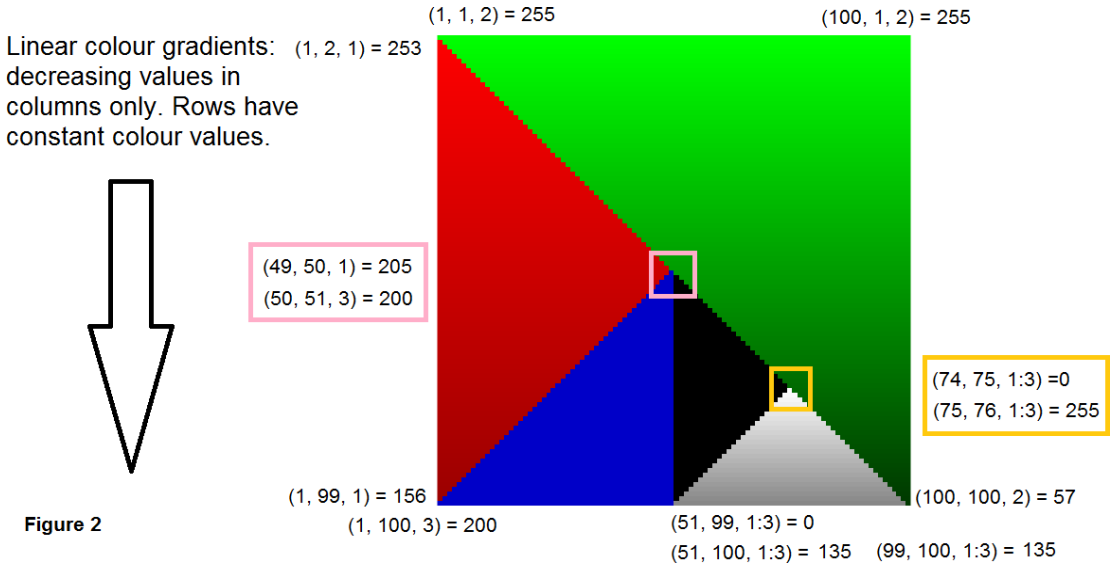


Figure 2

Summary of linear colour gradients

Black – no gradient all RGB values are 0 ie RGB [0 0 0]
Blue – no gradient, constant blue value of 200 RGB [0 0 200]
Green – decreases from RGB [0 255 0] in row 1 to [0 57 0] at pixel index (100, 100)
Red – decreases from RGB [253 0 0] at pixel index (1, 2) to [156 0 0] at index (1, 99)
Grey – decreases from [255 255 255] at index (75, 76) to [135 135 135] in row 100

2	***Add a few short comments concisely stating the important stages of your process. E.g., state the purpose of the conditional tests and what is happening in the code inside each conditional test.	1
	Total marks =	**7 %**

Now start task 3

Coursework Task 3

Step	Task 3 Instructions (* indicates your work will not be marked if this is not included) **Online gambling simulation: European Roulette**	Mark(s) %
1	Deactivate your entire block of code for task 2 using %{ and %}. Rename and save the provided T30000000.m file as instructed in the preliminary work section. Read the information in the file.	-
2	Copy and paste the code you created in lab worksheet 5 (LO 5.2) into the file from step 1 above.	-
3	A European Roulette wheel is divided into a number of red and black compartments plus one green compartment. When a gambler places a bet on 'Red' or 'Black', if the ball lands in the green compartment, it is counted as a win for the house. So, for example, if a gambler places a £5 bet on the ball landing in a red compartment and the ball lands in a black compartment *or* the green compartment, the house wins (the gambler loses £5.) Use Google to look up 'European Roulette wheel.' Look at the numerical layout of the wheel and calculate the probability that the house will win if a gambler bets on 'Red' (or if the gambler bets on 'Black') Now answer Matlab coursework task question **10**	1
4	Change the value for the probability in the 'backward step' condition in your program to the calculated probability (correct to 2 dp) of the house winning. Five different players subscribe to an online gambling site and play the roulette game. Each gambler has £200 to start with. The game ends when one of the following 3 conditions has been met (which ever condition is met first, the game ends) (**1**) a gambler has won £100 (**2**) the house has won £100 (i.e the gambler has lost £100) (**3**) a maximum of 20 plays are permitted, at which point the game ends and the gambler collects any winnings or pays out to the house. Modify your code to include the boundary conditions above. *Add a few short comments to indicate where your code has been modified. Run your code 5 times to simulate the 5 players games and each time, save the figure which is output. Paste the 5 figures into the area for Matlab coursework task question **11a-e**.	1 2 1 **1**
	Based on the simulation of the 5 players, how much money would you have made (or lost) so far from your online casino? Now answer Matlab coursework task question **12**	**1**
	Total marks =	**7 %**

Upload all of your work as a zipped folder (2 × matlabscript .m files – using the supplied template files plus 1 × Word document containing your answers to questions) on SurreyLearn *before* 16:00 on Wednesday 22nd November 2023.

Your submitted work and answers must be your own individual work