University of Wolverhampton
Faculty of Science and Engineering
Department of Mathematics and Computer Science

Module Assessment

Module	5CS021 – Numerical Methods and Concurrency
Module Leader	Hiran Patel
Semester	1
Year	2024/25
Assessment	Portfolio
% of module mark	100%
Due Date	See below
Hand-in – what?	**Portfolio as specified in this document**
Hand-in- where?	Canvas
Pass mark	40%
Method of retrieval	Submit the resit assessment (will be distributed after first grade has been released)
Feedback	Individual feedback via Canvas, in addition verbal feedback is available in class.
Collection of marked work	N/A

Assessment overview

This portfolio is split up into 4 separate tasks which will test your knowledge of mathematics, C programming and concurrency. Each task should be zipped up into a single zip folder containing all C and resource files for the submission on Canvas.

1. Linear Regression (20% - 100 marks)

This task will test your knowledge of file input and mathematical formulas. Basic linear regression (LR) is used to find a relationship between two types of data. For example, you could use LR to find the relationship between glucose intake and heart rate. These initially would be plotted on a graph where the x axis represents glucose, and the y axis represents heart rate. Using the LR formula, you can find y=bx+a (equation of a straight line (more commonly known as y=mx+c) between “n” number of points on a graph. Below are the formulas to find “a” and “b.”

$$A = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$B = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

You will be given multiple text files containing “n” number of coordinates. Your program will read in the coordinates (x,y) and use the LR formula to produce the gradient (b) and the constant (a), and being able to print out y=bx+a. For example, if a = 0.5 and b=-2, your program will print out y=-2x+0.5. This equation now represents the trend in the data you will be given. Finally, your program will ask the user to type in a value of x which will then calculate y.

NOTE – Your program only needs to take in one file. We have given you multiple files as they contain a different amount of numbers to test your program with.

Read data from file appropriately (30 marks)

Use LR formula to process the data and print out the correct equation of a straight line (50 marks)

Find new value of “y” using user input (20 marks)

2. Calculating Pi using Leibniz formula and multithreading (20% - 100 marks)

The Leibniz formula is an infinite series method of calculating Pi. The formula is a very simple way of calculating Pi, however, it takes a large amount of iterations to produce a low precision value of Pi. This task requires a large amount of computation and therefore it is vital that you use multithreading to speed up the program. Below is the Leibniz formula:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}.$$

As the series can be iterated infinite amount of times, your program should allow the user to take in 2 command line arguments; the first is the number of iterations (argv[1]) and the second is the amount of threads (argv[2]) the user would like to use. This means that the slicing of workload needs to be dynamic. NOTE – this program should work with any amount of threads.

Calculating Pi using Leibniz formula (20 marks)

Using multithreading with appropriate slicing (60 marks)

Correct value of Pi printed out depending on iteration count (20 marks)

3. Creating derivatives and finding gradients of a point using x from a list of functions within a text file using multithreading (30% - 100 marks)

You will be given multiple text files containing a list of functions (y = f(x)). On the end of each function, you will be given an “x” value which will need to input into the derivative function to find the gradient of the polynomial at a given point. The number of lines within each text file is random so your program should work with a text file with any amount of data (you will be given 4 to test with). You will create a C program which reads in the file appropriately, find dy/dx per polynomial, input “x” into the derivative and return the answer. You should then print the derivative function and the result given when substituting the x value into a separate text file. The aim of this task is to use POSIX threads to parallelise the task to take advantage of the multicore processor within your

machine to speed up the program. The threads you spawn within the program must compute an equal or close to an equal number of computations to make the program more efficient. You will determine the number of threads using argv[1] to allow the user to input this when executing the program. This question also requires you to use dynamic memory allocation.

NOTE – this program should work with any number of threads

NOTE – Your program only needs to take in one file. We have given you multiple files as they contain a different amount of numbers to test your program with.

An example of finding the derivative and substituting x. If you are given y = 10x^3 + 4x^4 + 3x (x=9), dy/dx will be 30x^2 + 16x^3 + 3. You then substitute the x value; in this case it is 9. 30(9)^3 + 16(9)^3 + 3 = 14097. You will then print out 30x^2 + 16x^3 + 3 as the derivative and next to it, the gradient value which is 14097. You may wish to do this by using m = [value]. If you would like to order based on the exponent value and the print to a file, you can but it is not required.

File read appropriately and data stored correctly (formulas) (10 marks)

Using dynamic memory – "malloc" (10 marks)

Creating functionality to find dy/dx of formulas read in from the file (20 marks)

Substituting x correctly into the derivative to find the gradient at a given point (10 marks)

Using multithreading with equal computations (40 marks)

Outputting correct output to a file (10 marks)

4. Box Blur with multithreading (30% - 100 marks)

Your program will decode a PNG file into an array and apply the box blur filter. Blurring an image reduces noise by taking the average RGB values around a specific pixel and setting it's RGB to the mean values you've just calculated. This smoothens the colour across a matrix of pixels. For this assessment, you will use a 3x3 matrix. For example, if you have a 5x5 image such as the following (be aware that the coordinate values will depend on how you format your 2D array):

0,4	1,4	2,4	3,4	4,4
0,3	1,3	2,3	3,3	4,3
0,2	1,2	2,2	3,2	4,2
0,1	1,1	2,1	3,1	4,1
0,0	1,0	2,0	3,0	4,0

The shaded region above represents the pixel we want to blur, in this case, we are focusing on pixel 1,2 (x,y) (Centre of the matrix). to apply the blur for this pixel, you would sum all the **Red** values from the surrounding coordinates including 1,2 (total of 9 R values) and find the average (divide by 9). This is now the new Red value for coordinate 1,2. You must then repeat this for Green and Blue values. This must be repeated throughout the image. If you are working on a pixel which is not fully surrounded by pixels (8 pixels), you must take the average of however many neighbouring pixels there are.
NOTE – this program should work with any amount of threads.

Reading in an image file into a single or 2D array (10 marks)

Applying Box filter on image (20 marks)

Using multithreading appropriately to apply Box filter (40 marks)

Using dynamic memory – malloc (10 marks)

Outputting the correct image with Box Blur applied (20 marks)

Important Message

You may be asked to clarify your assessment after moderation has taken place. This is to ensure the work has been completed by the student.

You must achieve 40 percent overall to pass this module. There will be a resit opportunity during resit week (July) to achieve a pass.

Submission of work

Your completed work for assignments must be handed in on or before the due date. ***You must keep a copy or backup of any assessed work that you submit. Failure to do so may result in your having to repeat that piece of work.***

Penalties for late submission of coursework

Standard Faculty of Science and Technology arrangements apply.
ANY late submission (without valid cause) will result in 0 marks being allocated to the coursework.

Procedure for requesting extensions

If you have a valid reason for requiring an extension you must request an extension using e:vision.
Requests for extension to assignment deadlines should normally be submitted at least one week before the submission deadline and may be granted for a maximum of seven days (one calendar week).

Retrieval of Failure

A pass of 40% or above must be obtained overall for the module (but not necessarily in each assessment task).
Where a student fails a module they have the right to attempt the failed assessment(s) once, at the next resit opportunity (normally July resit period). If a student fails assessment for a second time they have a right to repeat (i.e. RETAKE) the module.

NOTE: STUDENTS WHO DO NOT TAKE THEIR RESIT AT THE NEXT AVAILABLE RESIT OPPORTUNITY WILL BE REQUIRED TO REPEAT THE MODULE.

Mitigating Circumstances (also called Extenuating Circumstances).

If you are unable to meet a deadline or attend an examination, and you have a valid reason, then you will need to request via e:vision **Extenuating Circumstances.**

Feedback of assignments

You will be given feedback when you demonstrate your work.

You normally have **two working weeks** from the date you receive your grade and feedback to contact and discuss the matter with your lecturer. See the Student's Union advice page http://www.wolvesunion.org/adviceandsupport/ for more details.

Registration

Please ensure that you are registered on the module. You can check your module registrations via e:Vision You should see your personal tutor or the Student Support Officer if you are unsure about your programme of study. The fact that you are attending module classes does not mean that you are necessarily registered. A grade may not be given if you are not registered.

Cheating
Cheating is any attempt to gain unfair advantage by dishonest means and includes **plagiarism** and **collusion.** Cheating is a serious offence. You are advised to check the nature of each assessment. You must work individually unless it is a group assessment.

Cheating is defined as any attempt by a candidate to gain unfair advantage in an assessment by dishonest means, and includes e.g. all breaches of examination room rules, impersonating another candidate, falsifying data, and obtaining an examination paper in advance of its authorised release.

Plagiarism is defined as incorporating a significant amount of un-attributed direct quotation from, or un-attributed substantial paraphrasing of, the work of another.

Collusion occurs when two or more students collaborate to produce a piece of work to be submitted (in whole or part) for assessment and the work is presented as the work of one student alone.