ENGG1003 - Assessable Lab 2

Brenton Schulz

1 Instructions

This is an **open book** assessment task.

You have **2 hours** to complete the task.

Weighting: **5%** of your final grade.

You may get graded after having completed all tasks or demonstrate each section as it is completed. If you complete Task 2 correctly that will automatically award marks for Task 1.

2 Background and Code Template

The following tasks use the template below:

```c
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

int main() {
   float image[4][4][3];
   int x, y;

   // Don't change this srand() line!
   srand(0);

   for(y = 0; y < 4; y++) {
      for(x = 0; x < 4; x++) {
         image[x][y][0] = (float)rand()/(float)RAND_MAX;
         image[x][y][1] = (float)rand()/(float)RAND_MAX;
         image[x][y][2] = (float)rand()/(float)RAND_MAX;
      }
   }
   // ----------------------------------------
   // Write your code below this comment block
   // ----------------------------------------

   return 0;
}
```

The code declares a 3D array, `image[][][]`, and initialises it with random values. Because `srand(0)` is called the "random" numbers should be the same each time the program is executed. These values are for demonstration purposes only, you are to imagine that it is storing RGB pixel data for a 4x4 pixel image. The three array indices are the x-coordinate, the y-coordinate, and the colour (red at the 3rd dimension's index 0, green at index 1, and blue at index 2) of each pixel. The data in each element is the red, green, or blue colour intensity at a particular x-y coordinate.

Colour intensity values are floating point numbers between 0 and 1.

Task 1: 3D Array Processing - 3 Marks

Within the `main()` function, write code which *thresholds* each pixel's colour value to either black (`0.0 0.0 0.0`) or white (`1.0 1.0 1.0`).

Pixels where the red value (`image[x][y][0]`) is strictly larger than `0.5` should be set to white and all other pixels set to black.

Print the first two rows (ie: y = 0 and 1, each with x = 1 to 4) of image data to the console in the following format:

```
x y:  R G B
```

The code will produce different results on different platforms due to differences in the `rand()` implementation. When run in CodeBlocks under Windows, with `srand(0)` as the "seed", the correct output is below:

```
0 0: 0.000000 0.000000 0.000000
1 0: 0.000000 0.000000 0.000000
2 0: 0.000000 0.000000 0.000000
3 0: 1.000000 1.000000 1.000000
0 1: 0.000000 0.000000 0.000000
1 1: 0.000000 0.000000 0.000000
2 1: 0.000000 0.000000 0.000000
3 1: 1.000000 1.000000 1.000000
```

Running under Ubuntu 20.04 results in:

```
0 0: 1.000000 1.000000 1.000000
1 0: 1.000000 1.000000 1.000000
2 0: 0.000000 0.000000 0.000000
3 0: 1.000000 1.000000 1.000000
0 1: 0.000000 0.000000 0.000000
1 1: 1.000000 1.000000 1.000000
2 1: 0.000000 0.000000 0.000000
3 1: 0.000000 0.000000 0.000000
```

Marking Guide: You may be awarded one of the following:

- **2.5 Marks:** Your code correctly calculates the thresholding result each pixel value, allocates it to the correct elements(s), but does not correctly output the result for any reason (eg: too many or too few elements are printed, the output format is incorrect, the wrong elements are printed due to a row/column major loop error, etc).
- **3 Marks:** Your code correctly calculates the thresholded values, stores the results correctly, and prints the correct output values in the correct format.

Task 2: Functions with Pointer Arguments - 1 Mark

Modify your program so that the thresholding calculation is performed in a function. The function should accept pointers to three `floats` and modify the three array values directly.

The function prototype is:

```c
void pixelThreshold(float *r, float *g, float *b);
```

And it should be called with the syntax:

```c
pixelThreshold(&image[x][y][0], &image[x][y][1], &image[x][y][2]);
```

The program output should be identical to the previous task.

Task 3: File I/O - 1 Mark

Modify your program so that the program output (as specified in Task 1) is written to a file.

The filename should be `output.txt`.

Your program should output to the console *and* the file.