**Assignment 2 – Cellular Automaton**

**Group 13 – Team members:**

Yaseen Lahmami – 170025540

Aylin Ibryamov – 170017397

Oskar Jankowski – 170012865

To start this assignment we first had to figure out how rule 30 is generated. This took some time but eventually we figured the concept out and from then on we managed to generate all the rules. We did this by converting decimal to binary and using that to generate any rule from 0 to 255. To generate the rule we used a for loop and printed the numbers row by row. We thought this was the best way to approach this solution. Finally we focused our attention to the 2D cellular automata. We chose three automatons to work on - Brains Brain, Conway’s game of life and Langton’s ant. For each 2D automaton we used a different C programs to separate each simulation as these automatons have completely different rules. As for organising our files it was a bit of a mess to start with. We put everything in one C file but eventually we used header files and linked our program with different C files to make things more readable and modulated.

During the process of writing the assignment we faced variety of problems, first issue was to get our head the whole concept of cellular automaton, in theory we understood, code wise we were clueless. Yet after a while some of us watched a youtube tutorials and read online forums. We did not go far without facing another problem which was 2D arrays and their usage in the various 2D cellular automaton. The first issue with 2D arrays was passing them into methods as arguments. We didn’t know that we had to pass the size of the rows and columns in the 2D array in the function parameters. To solve this we defined the parameters (arr[ROWS][COLS]) or we used double pointers as 2D arrays. After we solved this problem it was pretty straight forward in programming our 2D automatons however we struggled getting Brian’s brain automaton done in time but we have included this in our assignment just to show where we went wrong. Files was another problem we faced as we didnt have any experience using the file functions in C. We kept trying and trying but the problem was saving the 1D cellular automaton to a text file in the form of different characters, the way we fixed this problem is by opening that text file in visual studio code. Later we came up with a solution to loop through the array and write the array single character at a time. This finally worked after some head scratching.

Due to our time management skills we couldn't finish all the requirements. The one extra requirement was generating a 3D torus. We felt that there wasn’t enough information to use to have some idea to generate a 3D torus and we felt like we had to use the graphics library which would require more time in learning. Generating the 3D torus was our least priority but it would of been great if we managed to figure it out.

In conclusion, we all feel that we could of done a bit more especially figuring how to do the 3D torus. We learnt a lot during this assignment for example, organising our code, the nature of the elementary cellular automatons and making use of different libraries such as unistd.h ( this allowed us to ‘animate’ our 2D cellular automatons using usleep() function which takes in time in microseconds). Finally we managed to finish most of the requirements and overall we are happy of what we managed to accomplish.