

## COMP281 Assignment 2 Report

### 1. Problem ID: 1081 Title: Game of Life

#### Solution description:

Besides the 2D char board used to record the status of each cell, another 2D integer board is created. Basically, the function of the integer board is to look up how many living neighbours of each cell and record the number in the corresponding positions.

```
.X...X
X.X.X.
X...X.
X..XX.
..XX.X
...X.X
```

Figure 1: initial status of the board

```
222221
241323
242533
133443
123462
013241
```

Figure 2: numbers of living neighbours for initial status

Figure 1 and Figure 2 show how this works. For each simulating step, the integer board of living neighbours will be generated first. Then based on the number of living neighbours and rules of 'Game of Life', the char board will be updated to next step.

Another difficult point is to deal with boundary cells:

(i-1, j-1)	(i-1, j)	(i, j+1)
(i, j-1)	(i, j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

Figure 3: Neighbours of cell (i, j)

In general situations, cell (i, j) does not always have 8 neighbours, it may have 3 or 5 neighbours, depending on its position. Suppose cell (i, j) is at top left corner, then (i-1, j-1) does not even exist. If the value of cell (i-1, j-1) is still scanned in, this may cause exception or dirty read. My solution uses function 'fmax' and 'fmin' in 'math.h' (return the max/min number between 2 numbers) for boundary detection. For every cell (i, j) in a  $m \times n$  board, the scanning range of its neighbours should be:

$$\text{Rows} \in [\max(0, i-1), \min(i+1, m-1)]$$

$$\text{Columns} \in [\max(0, j-1), \min(j+1, n-1)]$$

**2. Problem ID: 1086****Title: Run Length Encoding of ASCII Art****Solution description:**

A window of 2-character length is used to scan the input characters.

Previous input char	Current input char
------------------------	-----------------------

Figure 4: Scan window

- Compress an ASCII art file

Situation #	Window		Counter for char C	Operation Description
	Previous input char	Current input char		
1	C	C	cnt++	If 2 characters in the window are same, add 1 to its counter.
2	C	C'	cnt=1	Previous character appears only once, then print it directly.
3	C	C'	cnt>=2	Previous character appears 'cnt' times continuously, then print 'C C cnt *'. Meanwhile, set 'cnt' back to 1.
4	C	\n		Print a newline.

- Expand an ASCII art file

Situation #	Window		char buffer for output	Counter for char C	Operation Description
	Previous input char	Current input char			
1	C	C	NULL		As long as 2 characters in the window are same, set the char buffer to C and set counter to the number after.
2	C	C'	NULL		Previous character appears only once, then print it directly.
3	C	*	C	n	If current input char is a '*' and char buffer is not empty, then print char C for n times. Meanwhile, set char buffer back to NULL.
4	C	\n			Print a newline.

**Note: C' is a different character from C.**