# Northern Michigan University (Marquette Co, MI)
# CS 444-01-25F: Parallel And Distributed Processing

## Program 6
## Due: Wednesday 12 November 2025 9:00 A.M. EST

## The Game Of Life

Create a folder called "PG6" in the top level of your CS444-01-25F folder. Place all files pertaining to this assignment into the top level of your PG6 folder. Place a (possibly empty) file called "DONE" into this folder when you are ready to have your programs graded. The only files you need to turn in are the .cc and .h files. Please don't turn in any files other than these!!

Given an initial Game of Life grid, you are to compute and print the next *n* stages of the Game.

The Game of Life is played on a grid with *r* rows and *c* columns. A cell can be either alive or dead. A live cell remains alive in the next time step if it is adjacent (horizontally, vertically, or diagonally) to 2 or 3 other live cells; it dies otherwise. A dead cell becomes alive in the next time step if it is adjacent to exactly 3 live cells; it remains dead otherwise.

You will need *rc*+1 processors to simulate this; one processor for each cell, and one master process that will organize the data.

The master process will read the data from the input file and will send to each cell its initial configuration. At each time step, each cell will receive information from the surrounding cells to compute whether it should be dead or alive in the next time step and will send its status to the master process. The master process will receive the status from each cell and print out the complete grid at that time step.

The input file will be formatted as follows:
The first line will contain three integers separated by spaces: the number of rows, the number of columns, the number of time steps.
The next r rows will be the initial configuration of the board. A _ indicates a dead cell. A @ indicates a living cell.

For example,

```
4 4 3
___@
_@_@
@@@@
____
```

The output file will contain one grid for each time step. Do not print out the initial configuration.

In this case, the output would look something like: