Scott Chatham
schatham@ucsc.edu

This program implements a simple simulator for Conway's Game of Life with a 50x50 board.

build using %make
usage: ./GameofLife inputFile numGenerations

inputFile is the file used to read initial positions in and is required
numGenerations is an optional argument used to alter the number of generations simulated without having to make modifications to the input file

The program works by reading in initial positions from the given input file and then simulating the next generation using Update(). It keeps simulating until the given number of generations have been completed, after which it cleans up resources and exits.

Update() is where the bulk of the work is done and a high level overview of which follows. There are two distinct groups of cells in the program: live cells and dead neighbor cells. Each is represented by its own array. During each Update() call the following steps are taken:

1. find neighbors for all live cells and add them to the dead neighbor group
2. apply the game of life rules to dead neighbor cells by marking appropriate ones as alive
3. apply the game of life rules to the live cells
4. add live neighbor cells to live cells

The running order of Update() is O(n) where n is the number of live cells.
Step 1 takes roughly 8n steps as there are 8 locations we must check around a given cell to find the neighbors. Other work is also done such as adding cells or increasing the count of neighbors, but these are all a constant factor.
Step 2 takes roughly 8n because we are traversing the neighbor cells and applying rules.
Step 3 takes n steps because it simply traverses our live cells and checks which rule applies.
Step 4 again is roughly 8n as we check all the neighbor cells to see which ones become alive and add them to the live cells.