Programming Languages 3, 1. exercises:

1. Enter the Life program of this chapter on your computer and make sure that it works correctly.
2. Sometimes the user might wish to run the Life game on a grid smaller than 20 x 60. Determine how it is possible to make maxrow and maxcol into variables that the user can set when the program is run. Try to make as few changes in the program as possible.
3. One idea for speeding up the function Life :: neighbor_count(row, col) is to delete the hedge (the extra rows and columns that are always dead) fromthe arrays grid and new_grid. Then, when a cell is on the boundary, neighbor_count will look at fewer than the eight neighboring cells, since some of these are outside the bounds of the grid. To do this, the function will need to determine whether or not the cell (row, col) is on the boundary, but this can be done outside the nested loops, by determining, before the loops commence, the lower and upper bounds for the loops. If, for example, row is as small as allowed, Section 1.6 Conclusions and Preview 39 then the lower bound for the row loop is row; otherwise, it is row − 1. Determine, in terms of the size of the grid, approximately how many statements are executed by the original version of neighbor_count and by the new version. Are the changes proposed in this exercise worth making?
4. Modify the Life function initialize so that it sets up the initial Life :: grid configuration by accepting occupied positions as a sequence of blanks and x's in appropriate rows, rather than requiring the occupied positions to be entered as numerical coordinate pairs.
5. Add a feature to the function initialize so that it can, at the user's option, either read its initial configuration from the keyboard or from a file. The first line of the file will be a comment giving the name of the configuration. Each remaining line of the file will correspond to a row of the configuration. Each line will contain x in each living position and a blank in each dead position.
6. Add a feature to the Life program so that, at termination, it can write the final configuration to a file in a format that can be edited by the user and that can be read in to restart the program (using the feature of the previous exercise).
7. Add an animation feature to the game: in the beginning the game program asks a delay setting from the user. Then in starts running and it calculates new game situations and updates them to the screen with set time interval. For example if the user gave 2 seconds then the new situation is updated to the screen after two seconds continuously the the end of the game. You can terminate the game program with Ctrl-C or the game can ask from the user the number of updates it calculates.