Assignment 5
Play Minesweeper

Due: Sunday, March 13th, 2016 11:59pm

Grading: EVERY assignment in this course is graded by demoing your work for 10 minutes with a TA. You are required to meet with a TA within one week from the due date to demo. You are penalized for failure to see a TA within the week or missing a scheduled appointment. In either case, if you are within 1 day (24 hours) of the deadline, you lose 10 points. If you are within 7 days (1 week) of the deadline, then you lose 25 points, anything outside of a week from the deadline to demo is an automatic 50 point deduction. Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program☺

Implementation (70 points)
For this assignment, you will implement the game minesweeper. This game was made popular by the Windows operating system in the early 1990s, and it has continued to be a pre-installed game until Windows 8. You can watch a video of how to play minesweeper on YouTube or read about it on Wikipedia:
https://www.youtube.com/watch?v=Z0EAysRluJk
https://en.wikipedia.org/wiki/Microsoft_Minesweeper

The game is very simple. The object of the game is to open/reveal every cell on the board without detonating a mine. Every cell contains a number or a mine. The numbers tell you how many mines surround the cell in the horizontal, vertical, and diagonal directions, e.g. at most, 8 possible mines surrounding a cell. You can flag a cell, which helps you remember where you think there is a mine to not detonate. If you select to open/reveal a cell that has a mine, then you automatically lose the game!

Our game will be a little different, since we do not have a graphical user interface for clicking. **You will run the program providing command-line arguments for the number of rows, columns, and mines in any order.** Therefore, the number must be preceded by an option designator, -r for rows, -c for columns, and –m for mines.

You will then randomly distribute the mines on the board and setup the numbers describing how many adjacent mines to each cell. After that, you will display a blank board, and ask the user to flag or open a cell on the board until the user selects a cell that contains a mine (losing the game) or selects all cells free of mines (winning the game). After the user wins or loses, you will ask the user if he/she wants to play again. If so, you must get the number of rows, columns, and mines for the new game and create a new board for a new game. **BTW, you can display row and column numbers on board to make it easier to read/know which row and column to select!!!**

Example run of a game:

```
a.out –r 9 –c 9 –m 10
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------

Flag(1) or Open(2): 2
row, col: 0 4

You lose!!!
-------------------
|1|1|0|1|*|1|0|0|0|
-------------------
|*|2|1|2|2|2|0|0|0|
-------------------
|2|*|2|2|*|1|0|0|0|
-------------------
|1|1|3|*|3|1|0|0|0|
-------------------
|1|1|3|*|2|0|0|0|0|
-------------------
|1|*|3|2|2|0|0|0|0|
-------------------
|1|1|2|*|1|0|0|0|0|
-------------------
|1|2|3|2|1|0|0|0|0|
-------------------
|1|*|*|1|0|0|0|0|0|
-------------------

Do you want to play again (1-yes, 2-no): 1
How many rows, cols? 9 9
How many mines? 10
```

```
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------

Flag(1) or Open(2): 2
row, col: 0 0
-------------------
|1| | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------

Flag(1) or Open(2): 2
row, col: 8 0
-------------------
|1| | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
| | | | | | | | | |
-------------------
|1| | | | | | | | |
-------------------

Flag(1) or Open(2): 2
row, col: 5 5
-------------------
|1| | | | |1|0|0|0|
-------------------
| | | | | |2|0|0|0|
-------------------
| | | | | |1|0|0|0|
-------------------
| | | | |3|1|0|0|0|
-------------------
| | | | |2|0|0|0|0|
-------------------
| | | | |2|0|0|0|0|
-------------------
| | | | |1|0|0|0|0|
-------------------
| | | |2|1|0|0|0|0|
-------------------
|1| | |1|0|0|0|0|0|
-------------------

Flag(1) or Open(2): 1
row, col: 6 3
-------------------
|1| | | | |1|0|0|0|
-------------------
| | | | | |2|0|0|0|
-------------------
| | | | | |1|0|0|0|
-------------------
| | | | |3|1|0|0|0|
-------------------
| | | | |2|0|0|0|0|
-------------------
| | | | |2|0|0|0|0|
-------------------
| | | |!|1|0|0|0|0|
-------------------
| | | |2|1|0|0|0|0|
-------------------
|1| | |1|0|0|0|0|0|
-------------------
…


Flag(1) or Open(2): 2
row, col: 7 1
-------------------
|1|1|0|1|!|1|0|0|0|
-------------------
|!|2|1|2|2|2|0|0|0|
-------------------
|2|!|2|2|!|1|0|0|0|
-------------------
|1|1|3|!|3|1|0|0|0|
-------------------
|1|1|3|!|2|0|0|0|0|
-------------------
|1|!|3|2|2|0|0|0|0|
-------------------
|1|1|2|!|1|0|0|0|0|
-------------------
|1|2| |2|1|0|0|0|0|
-------------------
|1|!|!|1|0|0|0|0|0|
-------------------

Flag(1) or Open(2): 2
row, col: 7 2
Congratulations!!!
-------------------
|1|1|0|1|*|1|0|0|0|
-------------------
|*|2|1|2|2|2|0|0|0|
-------------------
|2|*|2|2|*|1|0|0|0|
-------------------
|1|1|3|*|3|1|0|0|0|
-------------------
|1|1|3|*|2|0|0|0|0|
-------------------
|1|*|3|2|2|0|0|0|0|
-------------------
|1|1|2|*|1|0|0|0|0|
-------------------
|1|2|3|2|1|0|0|0|0|
-------------------
|1|*|*|1|0|0|0|0|0|
-------------------

Do you want to play again (1-yes, 2-no): 2
```

Requirements for game:

- You must provide a usage message, if the user enters incorrect command-line arguments
- You must not have functions over 15-20 lines long (-10 automatically)
- You must not use global variables (-10 automatically)
- You must ask the user if he/she wants to play again
    - If no, then end
    - If yes, then prompt for rows, cols, and mines for new game.
- You must not have any memory leaks (-10 automatically)
- You must detect these errors:
    - Invalid row or column to flag or open
    - Opening a cell that has already been opened

(10 pts) Extra Credit: Recursively open all cells adjacent to 0 cell
If the user selects a cell that has no mines surrounding it, then you will help the user by recursively opening all adjacent cells to empty cells, until you reach cells with a surround mine.

(10 pts) Design

You will design a solution for the problem statement as well as answer the following questions for **each** function that you write.

- What is being passed into this function?
- What is being returned?
- What must be true **before** this function can be called?
- What will always be true **after** this function is called?

Your design can be in the form of a flow chart, pseudocode, or both. **DESIGN BEFORE YOU CODE.** While there is no way that we can check that you actually did, it will make writing the code take much less time.

(10 pts) Testing

Fill out this testing table for your program (the first row is an example and you must test more than one). Remember, you need to have good and bad input values, even if you don't handle the errors!!!

Inputted Values	Expected Output	Actual met
-1 4	`Invalid row`	yes
…		

(10 pts) Program Style/Comments
In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!
http://classes.engr.oregonstate.edu/eecs/winter2016/cs161-001/161_style_guideline.pdf

```
/******************************************************
** Program: mines.cpp
** Author: Your Name
** Date: 3/06/2016
** Description:
** Input:
** Output:
******************************************************/
```

Electronically submit your C++ program (.cpp file, not your executable!!!) and pdf, by the assignment due date, using TEACH.

https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth

**NOTE: The easiest way to upload your program from ENGR to TEACH is to map a network drive to your home directory on ENGR. Mac or Windows, See:

http://engineering.oregonstate.edu/computing/fileaccess/

If you are doing this off campus, pay attention to the off-campus directions!!!!