Final

Files to submit: **Makefile, all necessary .c and .h files need to compile your program, Readme.txt**

- All programs must compile without warnings when using the -Wall option
- If you are working in a group **ALL** members must submit the assignment on SmartSite
- Submit only the files requested
  - Do **NOT** submit folders or compressed files such as .zip, .rar, .tar, .targz, etc
- All output must match the provided solution in order to receive credit
  - We use a program to test your code so it must match exactly to receive credit
- All input will be valid unless stated otherwise
- The examples provided in the prompts do not represent all possible input you can receive. Please see the Tests folder for each problem for more adequate testing
- You may assume all inputs are valid unless otherwise specified
- All inputs in the examples in the prompt are underlined
- If you have questions please post them to Piazza

Restrictions
- No global variables are allowed
- Your main function may only declare variables and call other functions.

Testing
- Due to the use of random numbers you will have to test your code on the CSIF computers to see if it is correct.
  - I used pc 10 and it matched my laptop
  - You may also be able to use just a linux system and have it work but I don't know.

1. (Time: 2 – 3 hours) For your final project you will be implementing the computer game minesweeper. The game of minesweeper works as follows:
   1. There is a rectangular grid of tiles
   2. Hidden under some of the tiles are mines
   3. Your goal is to mark all the tiles that contain mines and uncover all the tiles that don't contain mines
   4. If you uncover a tile that contains a mine you lose the game
   5. If you uncover a tile that doesn't contain a mine, it tells you how many mines are hidden in the eight surrounding squares

Here are some example boards with all of their contents revealed so that you can get an idea as to what they look like.

Example 1:
```
2  *  *  2  0
1  5  *  3  0
0  *  *  2  0
   0  1  2  3
```

Example 2:
```
9 0 0 0 0 0 1 1 1 0 0
8 0 0 0 0 0 1 * 1 0 0
7 0 1 1 1 0 1 1 2 1 1
6 0 1 * 1 1 1 1 1 * 1
5 0 1 1 1 1 * 1 1 1 1
4 1 1 1 0 1 1 1 0 0 0
3 1 * 1 0 0 0 0 0 0 0
2 1 1 1 0 0 0 0 0 0 0
1 1 2 1 1 0 1 1 2 2 2
0 * 2 * 1 0 1 * 2 * *
  0 1 2 3 4 5 6 7 8 9
```

Requirements

1. Name your executable **mine_sweeper.out**

2. You must submit a makefile named **Makefile,** along with the necessary .c and .h files needed to compile your program
   1. You must have at least 2 .c files

3. Your program should accept the following command line arguments
   1. The number of rows

2. The number of columns

3. The number of mines

4. Optionally a seed to the random number generator

   1. If no seed is given then you should seed the random number generator with the current time

5. If not enough command line parameters are given your program should print a usage message and terminate.

4. You **MUST USE A STRUCT** to store your board state

   1. This includes but is not limited to: the size of your board, the number of mines on the board, and the status of each tile (revealed, concealed, marked, etc)

5. During each turn the user will be asked to select a row and column number to take an action on

   1. You must validate that the tile they select is contained within the board

   2. If the row and column are not valid, you should repeatedly ask the user for a new row and column until they enter a valid one.

6. After selecting a tile the user will be presented with a list of actions to take based on the tile they selected

   1. If the tile is **concealed**, they may reveal it, mark it as a mine, mark it as a possible mine, or cancel their move

   2. If the tile is **marked as a mine,** they may remove their marker or cancel their move.

   3. If the tile is **marked as a possible mine,** they may remove their marker or cancel their move

   4. If the tile is already revealed tell the user that the tile is already revealed

   5. If the user chooses to cancel their action or the tile is already revealed the user should be asked for a new tile and a new action to take on it

7. We will be using the following symbols in your board

   1. #: a concealed tile

   2. *: a mine

   3. !: a tile marked as a mine

   4. ?: a tile marked as potentially being a mine

8. Since mines will be randomly distributed around the board it is important that you follow the same algorithm for placing mines as I did. The algorithm is

1. For each mine that needs to be placed:
    1. randomly choose a row
    2. randomly choose a column
    3. If the board at the row and column does not contain a mine place it there
    4. else repeat beginning at sub-step 1
2. You should also print out where you place your mines so that you can confirm that they are being placed in the same locations as my program
9. Each tile that does not contain a mine, once revealed, shows the number of mines contained in the 8 tiles surrounding it
    1. If the user reveals a concealed tile that has 0 mines surrounding it, then not only that tile but all concealed tiles that do not contain mines next to it should also be revealed
        1. If a new tile is encountered that has 0 mines surrounding it, the process repeats
        2. Hint, this involves recursion
10. The user losers if they reveal a mine
11. The user wins if they reveal all non-mine tiles **and** have marked all tiles that contain a mine
12. After the game is over the entire board should be revealed whether the user won or not should be displayed.

Example 1:
```
./mine_sweeper.out 4 5 2 2
Placing mine at 2, 4
Placing mine at 0, 0
There are 2 mines left
3 # # # # #
2 # # # # #
1 # # # # #
0 # # # # #
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 3 0
Enter Action
```

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 0

There are 2 mines left

3 0 0 0 1 #

2 0 0 0 1 #

1 1 1 0 1 1

0 # 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 0 0

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 2

There are 1 mines left

3 0 0 0 1 #

2 0 0 0 1 #

1 1 1 0 1 1

0 ! 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 2 4

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 1

There are 1 mines left

3 0 0 0 1 #

2 0 0 0 1 ?

1 1 1 0 1 1

0 ! 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: <u>3 4</u>

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: <u>0</u>

There are 1 mines left

3 0 0 0 1 1

2 0 0 0 1 ?

1 1 1 0 1 1

0 ! 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: <u>2 4</u>

Enter Action

0. UnQuestion

1. Cancel

Action: <u>0</u>

There are 1 mines left

3 0 0 0 1 1

2 0 0 0 1 #

1 1 1 0 1 1

0 ! 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: <u>0 0</u>

Enter Action

0. UnMark

1. Cancel

Action: 0

There are 2 mines left

3 0 0 0 1 1

2 0 0 0 1 #

1 1 1 0 1 1

0 # 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 0 0

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 2

There are 1 mines left

3 0 0 0 1 1

2 0 0 0 1 #

1 1 1 0 1 1

0 ! 1 0 0 0

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 1 1

This tile is already revealed.

Enter row a row between 0-3 and a column between 0-4: 0 2

This tile is already revealed.

Enter row a row between 0-3 and a column between 0-4: 2 99

Enter row a row between 0-3 and a column between 0-4: 2 4

Enter Action

0. Reveal

```
1. Question
2. Mark
3. Cancel
Action: 0
3 0 0 0 1 1
2 0 0 0 1 *
1 1 1 0 1 1
0 * 1 0 0 0
  0 1 2 3 4
You Lost :(
```

Example 2:

```
./mine_sweeper.out 4 5 3 2
Placing mine at 2, 4
Placing mine at 0, 0
Placing mine at 1, 3
There are 3 mines left
3 # # # # #
2 # # # # #
1 # # # # #
0 # # # # #
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 2 2
Enter Action
0. Reveal
1. Question
2. Mark
3. Cancel
Action: 1
There are 3 mines left
```

```
3 # # # # #
2 # # ? # #
1 # # # # #
0 # # # # #
  0 1 2 3 4
```
Enter row a row between 0-3 and a column between 0-4: <u>0 4</u>

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: <u>0</u>

There are 3 mines left

```
3 # # # # #
2 # # ? # #
1 # # # # #
0 # # # # 1
  0 1 2 3 4
```
Enter row a row between 0-3 and a column between 0-4: <u>3 0</u>

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: <u>0</u>

There are 3 mines left

```
3 0 0 0 1 #
2 0 0 ? 2 #
1 1 1 1 # #
0 # # # # 1
  0 1 2 3 4
```

Enter row a row between 0-3 and a column between 0-4: 0 1

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 0

There are 3 mines left

3 0 0 0 1 #

2 0 0 ? 2 #

1 1 1 1 # #

0 # 1 # # 1

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 0 0

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 2

There are 2 mines left

3 0 0 0 1 #

2 0 0 ? 2 #

1 1 1 1 # #

0 ! 1 # # 1

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 2 2

Enter Action

0. UnQuestion

1. Cancel

Action: 0

```
There are 2 mines left
3 0 0 0 1 #
2 0 0 # 2 #
1 1 1 1 # #
0 ! 1 # # 1
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 2 2
Enter Action
0. Reveal
1. Question
2. Mark
3. Cancel
Action: 0
There are 2 mines left
3 0 0 0 1 #
2 0 0 1 2 #
1 1 1 1 # #
0 ! 1 # # 1
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 3 1
This tile is already revealed.
Enter row a row between 0-3 and a column between 0-4: 1 3
Enter Action
0. Reveal
1. Question
2. Mark
3. Cancel
Action: 2
There are 1 mines left
3 0 0 0 1 #
2 0 0 1 2 #
```

```
1 1 1 1 ! #
0 ! 1 # # 1
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 1 4
Enter Action
0. Reveal
1. Question
2. Mark
3. Cancel
Action: 0
There are 1 mines left
3 0 0 0 1 #
2 0 0 1 2 #
1 1 1 1 ! 2
0 ! 1 # # 1
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 0 2
Enter Action
0. Reveal
1. Question
2. Mark
3. Cancel
Action: 0
There are 1 mines left
3 0 0 0 1 #
2 0 0 1 2 #
1 1 1 1 ! 2
0 ! 1 1 # 1
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 0 3
Enter Action
```

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 0

There are 1 mines left

3 0 0 0 1 #

2 0 0 1 2 #

1 1 1 1 ! 2

0 ! 1 1 1 1

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 2 4

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 2

There are 0 mines left

3 0 0 0 1 #

2 0 0 1 2 !

1 1 1 1 ! 2

0 ! 1 1 1 1

  0 1 2 3 4

Enter row a row between 0-3 and a column between 0-4: 3 4

Enter Action

0. Reveal

1. Question

2. Mark

3. Cancel

Action: 0

```
3 0 0 0 1 1
2 0 0 1 2 *
1 1 1 1 * 2
0 * 1 1 1 1
  0 1 2 3 4
You Won!!
```

Example 3:

```
./mine_sweeper.out 4 5 3 2
Placing mine at 2, 4
Placing mine at 0, 0
Placing mine at 1, 3
There are 3 mines left
3 # # # # #
2 # # # # #
1 # # # # #
0 # # # # #
  0 1 2 3 4
Enter row a row between 0-3 and a column between 0-4: 0 0
Enter Action
0. Reveal
1. Question
2. Mark
3. Cancel
Action: 0
3 0 0 0 1 1
2 0 0 1 2 *
1 1 1 1 * 2
0 * 1 1 1 1
  0 1 2 3 4
```

You Lost :(