

Assignment 6: Lists and Arrays

[1] **Objectives:** This assignment aims to practice using lists and arrays, defining functions, passing parameters, and return values. What is more challenging for some students is their problem-solving ability. We have worked on linear lists in many examples before, so that we will concentrate on lists of lists or 2-dimensional arrays in this assignment.

[2] **Description:** You are supposed to implement a simple minesweeper game with one player. Minesweeper rules are straightforward. The board is divided into cells, with mines randomly distributed (one per cell). To win, you need to open all the cells without a mine. If you open a cell with a mine, the game will end. Some games will hint at the proximity of the mines in a neighborhood. We will keep it simple, so you depend on luck. A description can be found at <https://minesweeper.online/>. Remember, the game there is more complex than ours. The figure is taken from the website. The number of a cell is the number of mines in the neighboring cells. You can try the more complicated version on your own.



Here are some parameters of the game.

- A map is divided into n by n (square) grids with index (i, j) pinpointing to a cell.
- The player will be prompted for the number of mines on the map before the game starts. This number must be two or higher. So, the game will be over eventually.
- The player opens (“lands on”) one cell at a time. You can continue to play if there is no mine there. Since we cannot click on a cell, we will ask the player for a pair of indices (row and column).
- A player will get 10 points for each cell opened.
- Keep two maps. The first one is a MineMap which remembers the locations of the mines. This map will not change after the initial setup phase. An at-sign (@) denotes a cell with a mine.
- The second map is a GameMap which traces the players opened cells. A dot (.) denotes a safely opened cell, and an asterisk (*) denotes an opened cell with a mine. This map will be updated for each iteration.

Generally, it is a bad idea to use global variables. However, if a variable is used as a constant, i.e., it never changes the value, we sometimes use global variables to simplify the code. In this program, you can use size, blank, @-sign, and *-sign as global variables.

Some helpful suggestions:

- Use a fixed seed for a random number generator to test the algorithm with the same data.
- Start the test with the map size equal to 5 x 5 first. Assume the map is more than 5x5 and no more than 20 x 20.
- You may want to print out the MineMap when debugging (as shown in the screenshot below).

[3] **Requirements:** Your program should have at least the following functions.

- PrintMap(map): This function takes a map and prints it out nicely. It should work for both maps mentioned above.

- HideMines(map): This is used to initialize the MineMap.
- Play(MineMap, GameMap): This function contains a loop that prompts the player to open a (not-yet-opened) cell and decide what to do next.

[4] **Output:** A sample output is given below. You should test for all possibilities.

[5] **Deadline:** 11:59 pm, Monday, March 27, 2023

```

The number of bombs to set for the game: 5
There are 5 bomb(s) hidden. Good luck!
  0  1  2  3  4
0 [ ] [ ] [ @ ] [ ] [ @ ]
1 [ ] [ ] [ ] [ ] [ ]
2 [ ] [ @ ] [ ] [ ] [ ]
3 [ @ ] [ ] [ ] [ ] [ ]
4 [ @ ] [ ] [ ] [ ] [ ]

Enter the row: 1
Enter the column: 0

Game Board:
  0  1  2  3  4
0 [ ] [ ] [ ] [ ] [ ]
1 [ . ] [ ] [ ] [ ] [ ]
2 [ ] [ ] [ ] [ ] [ ]
3 [ ] [ ] [ ] [ ] [ ]
4 [ ] [ ] [ ] [ ] [ ]

Enter the row: 1
Enter the column: 1

Game Board:
  0  1  2  3  4
0 [ ] [ ] [ ] [ ] [ ]
1 [ . ] [ . ] [ ] [ ] [ ]
2 [ ] [ ] [ ] [ ] [ ]
3 [ ] [ ] [ ] [ ] [ ]
4 [ ] [ ] [ ] [ ] [ ]

Enter the row: 1
Enter the column: 2

Game Board:
  0  1  2  3  4
0 [ ] [ ] [ ] [ ] [ ]
1 [ . ] [ . ] [ . ] [ ] [ ]
2 [ ] [ ] [ ] [ ] [ ]
3 [ ] [ ] [ ] [ ] [ ]
4 [ ] [ ] [ ] [ ] [ ]

Enter the row: 0
Enter the column: 2

*** Game Over. Your Score is 40.

  0  1  2  3  4
0 [ ] [ ] [ * ] [ ] [ ]
1 [ . ] [ . ] [ . ] [ ] [ ]
2 [ ] [ ] [ ] [ ] [ ]
3 [ ] [ ] [ ] [ ] [ ]
4 [ ] [ ] [ ] [ ] [ ]

*** Here are all the bomb(s):
  0  1  2  3  4
0 [ ] [ ] [ @ ] [ ] [ @ ]
1 [ ] [ ] [ ] [ ] [ ]
2 [ ] [ @ ] [ ] [ ] [ ]
3 [ @ ] [ ] [ ] [ ] [ ]
4 [ @ ] [ ] [ ] [ ] [ ]

```