PseudoCode
1. Initialize the game board with empty cells and randomly place mines.
2. Print the initial state of the board to the player.
3. Ask the player to select a cell to reveal.
4. If the selected cell contains a mine, end the game.
5. If the selected cell is empty, reveal it and recursively reveal
neighboring cells until all adjacent cells are revealed.
6. Update the board to reflect the changes made by the player.
7. Repeat steps 3-6 until the game is over.
8. If the game is over, reveal all remaining cells to the player and
display the final state of the board.
9. If all non-mine cells are revealed, the player wins the game.
10. End the game.

RUNS

```
   0 1 2 3 4 5 6 7 8 9
0  # # # # # # # # # #
1  # # # # # # # # # #
2  # # # # # # # # # #
3  # # # # # # # # # #
4  # # # # # # # # # #
5  # # # # # # # # # #
6  # # # # # # # # # #
7  # # # # # # # # # #
8  # # # # # # # # # #
9  # # # # # # # # # #
Enter row and column (e.g., 3 4): 0 0
   0 1 2 3 4 5 6 7 8 9
0  0 0 0 0 0 0 1 # # # #
1  1 1 1 0 0 0 1 # # # #
2  # 1 0 1 1 2 # # # #
3  1 1 1 0 1 # # # # # #
4  0 0 0 0 1 2 # # # # #
5  1 1 0 0 1 1 1 1 # #
6  # 1 0 0 0 0 0 1 # #
7  # 2 0 0 1 1 1 1 # #
8  # 1 0 0 1 # # # # #
```

```
Enter row and column (e.g., 3 4): 0 9
   0 1 2 3 4 5 6 7 8 9
0  0 0 0 0 0 0 1 # # # 1
1  1 1 1 0 0 0 1 # # # #
2  # 1 0 1 1 2 # # # #
3  1 1 1 0 1 # # # # # #
4  0 0 0 0 1 2 # # # # #
5  1 1 0 0 1 1 1 1 # #
6  # 1 0 0 0 0 0 1 # #
7  # 2 0 0 1 1 1 1 # #
8  # 1 0 0 1 # # # # #
9  # 1 0 0 1 # # # # #
Enter row and column (e.g., 3 4): 3 0
   0 1 2 3 4 5 6 7 8 9
0  0 0 0 0 0 0 1 # # # 1
1  1 1 1 0 0 0 1 # # # #
2  # 1 0 1 1 2 # # # #
3  1 1 1 0 1 # # # # # #
4  0 0 0 0 1 2 # # # # #
```

```
3  1 1 1 0 1 # # # # # #
4  0 0 0 0 1 2 # # # # #
5  1 1 0 0 1 1 1 1 # #
6  # 1 0 0 0 0 0 1 # #
7  # 2 0 0 1 1 1 1 2 2
8  # 1 0 0 1 # 1 0 0 0
9  1 1 0 0 1 # 1 0 0 0
Enter row and column (e.g., 3 4): 8 0
   0 1 2 3 4 5 6 7 8 9
0  0 0 0 0 0 0 1 1 1 1 1
1  1 1 1 0 0 0 1 M # # M
2  M 1 0 1 1 2 # # # #
3  1 1 1 0 1 M # # # # #
4  0 0 0 0 1 2 M # # # #
5  1 1 0 0 1 1 1 1 # #
6  M 1 0 0 0 0 0 1 M M
7  # 2 0 0 1 1 1 1 2 2
8  M 1 0 0 1 M 1 0 0 0
9  1 1 0 0 1 # 1 0 0 0
Game Over!
```

```
10 have been placed around randomly.
   0 1 2 3 4 5 6 7 8 9
0 # # # # # # # # # #
1 # # # # # # # # # #
2 # # # # # # # # # #
3 # # # # # # # # # #
4 # # # # # # # # # #
5 # # # # # # # # # #
6 # # # # # # # # # #
7 # # # # # # # # # #
8 # # # # # # # # # #
9 # # # # # # # # # #
Enter row and column (e.g., 3 4): 2 3
   0 1 2 3 4 5 6 7 8 9
0 # # # # # # # # # #
1 # # # # # # # # # #
2 # # # 1 # # # # # #
3 # # # # # # # # # #
4 # # # # # # # # # #
5 # # # # # # # # # #
6 # # # # # # # # # #
7 # # # # # # # # # #
8 # # # # # # # # # #
9 # # # # # # # # # #
Enter row and column (e.g., 3 4): e 1
Please enter valid numbers!
```

```
   0 1 2 3 4 5 6 7 8 9
0 0 0 2 M 2 0 0 1 # #
1 0 0 2 M 2 0 0 1 M #
2 0 0 1 1 1 0 0 1 1 1
3 0 0 0 0 0 0 0 0 0 0
4 0 1 1 1 0 0 0 0 0 0
5 0 1 M 2 2 1 1 1 2 2
6 0 1 2 M # M # # M M
7 0 0 1 1 2 # # # # #
8 0 0 0 0 1 # # # # M
9 0 0 0 0 1 M # # # 1
Game Over!
```

```
   0 1 2 3 4 5 6 7 8 9
0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 1 1 0 0
2 0 0 0 0 1 2 # 1 0 0
3 0 0 0 0 1 # 2 1 0 0
4 0 0 0 0 1 1 1 0 0 0
5 1 1 0 0 0 0 1 1 1 0
6 # 2 2 2 1 0 1 # 1 0
7 # # # # 1 0 1 1 1 0
8 # # # # 2 2 2 1 0 0
9 1 # # # # # 1 0 0
Enter row and column (e.g., 3 4): 8 5e
Please enter valid numbers!
8 5
   0 1 2 3 4 5 6 7 8 9
0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 1 1 0 0
2 0 0 0 0 1 2 # 1 0 0
3 0 0 0 0 1 # 2 1 0 0
4 0 0 0 0 1 1 1 0 0 0
5 1 1 0 0 0 0 1 1 1 0
6 # 2 2 2 1 0 1 # 1 0
7 # # # # 1 0 1 1 1 0
8 # # # # 2 2 2 1 0 0
9 1 # # # # # 1 0 0
Enter row and column (e.g., 3 4):
```

ERRORS/DEBUGS

```
Exception in thread "main" java.util.InputMismatchException  Create breakpoint
	at java.base/java.util.Scanner.throwFor(Scanner.java:943)
	at java.base/java.util.Scanner.next(Scanner.java:1598)
	at java.base/java.util.Scanner.nextInt(Scanner.java:2263)
	at java.base/java.util.Scanner.nextInt(Scanner.java:2217)
	at Minesweeper.playGame(Minesweeper.java:95)
	at TestMinesweeper.main(TestMinesweeper.java:5)
```

```
  0 1 2 3 4 5 6 7 8 9
0 # # # # # # # # # #
1 # # # # # # # # # #
2 # # # # # # # # # #
3 # # # # # # # # # #
4 # # # # # # # # # #
5 # # # # # # # # # #
6 # # # # # # # # # #
7 # # # # # # # # # #
8 # # # # # # # # # #
9 # # # # # # # # # #
Enter row and column (e.g., 3 4): 100 6
  0 1 2 3 4 5 6 7 8 9
0 # # # # # # # # # #
1 # # # # # # # # # #
2 # # # # # # # # # #
3 # # # # # # # # # #
4 # # # # # # # # # #
5 # # # # # # # # # #
6 # # # # # # # # # #
7 # # # # # # # # # #
8 # # # # # # # # # #
9 # # # # # # # # # #
Enter row and column (e.g., 3 4): |
```

Showing it wont go out of bounds

```
    revealCell(row, col);

    if (!gameOver) {
        checkWin();
    }
        Cannot resolve method 'checkWin' in 'Minesweeper'
```

Forgot a "}"

```
public static char[][] getVisible() {
    return visible;
}
```

Forgot "}"

```java
no usages
public boolean isGameOver() {
    return gameOer;
}
```

TYPO!