AP 2003 Question 3

A treasure map is represented as a rectangular grid. Each grid location contains either a single treasure or nothing. The grid is represented using a two-dimensional array of boolean values. If a cell in the grid contains a treasure then the value true if stored in the corresponding array location. Otherwise, the value false is stored.

private boolean [][] treasureMap;

1. Write the TreasureMap method hasTreasure which returns true if there is a treasure at the location (row, column). If (row, column) is not within the bounds of the grid or if there is no treasure at that location, hasTreasure returns false.

public boolean hasTreasure (int row, int column)

1. Write the TreasureMap method numAdjacent, which returns the number of treasures that are located in locations adjacent to the given specified by row and column. To be adjacent, a treasure must be in one of the (at most) eight cells that border the given location horizontally, vertically or diagonally. A treasure in the given location does not count as being adjacent.

For example, consider the following TreasuureMap, with treasures represented by the letter X:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|   | X | X |   | X |   | X |   |   |
|   | X |   |   |   |   | X |   |   |
|   | X |   | X | X |   |   | X | X |
| X |   | X |   | X | X |   |   |   |
|   | X |   |   | X |   |   | X |   |
| X |   |   | X |   | X |   |   |   |

If theMap represents the above treasure map, the following table gives the results of several calls to numAdjacent:

|  |  |
| --- | --- |
| Method Call | Value Returned |
| theMap.numAdjacent(3,3) | 5 |
| theMap.numAdjacent(2,4) | 3 |
| theMap.numAdjacent(4,7) | 0 |

public int numAdjacent(int r, int c)

c) The method computeCounts returns the matrix of integers where the value at (row, column) is 9 if there is a treasure in location (row, column) in theMap. Otherwise, the value at (row, column) is the number of treasures adjacent to location (row, column).

For example, the following table shows an example of a TreasureMap object (aMap) and the array returned by computeCounts:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |
| --- |
| aMap |
|   | X |   | X | X |
| X |   |   |   |   |
|   | X | X |   |   |

 |

|  |
| --- |
| Returned by computeCounts |
| 2 | 9 | 2 | 9 | 9 |
| 9 | 4 | 4 | 3 | 2 |
| 2 | 9 | 9 | 1 | 0 |

 |

Implement the computeCounts method.

public int[][] computerCounts()