

4. This question involves reasoning about a number puzzle that is represented as a two-dimensional array of integers. Each element of the array initially contains a value between 1 and 9, inclusive. Solving the puzzle involves clearing pairs of array elements by setting them to 0. Two elements can be cleared if their values sum to 10 or if they have the same value. The puzzle is considered solved if all elements of the array are cleared.

You will write the constructor and one method of the `SumOrSameGame` class, which contains the methods that manipulate elements of the puzzle.

```
public class SumOrSameGame
{
    private int[][] puzzle;

    /**
     * Creates a two-dimensional array and fills it with random integers,
     * as described in part (a)
     * Precondition: numRows > 0; numCols > 0
     */
    public SumOrSameGame(int numRows, int numCols)
    { /* to be implemented in part (a) */ }

    /**
     * Identifies and clears an element of puzzle that can be paired with
     * the element at the given row and column, as described in part (b)
     * Preconditions: row and col are valid row and column indices in puzzle.
     * The element at the given row and column is between 1 and 9, inclusive.
     */
    public boolean clearPair(int row, int col)
    { /* to be implemented in part (b) */ }

    /* There may be instance variables, constructors,
       and methods that are not shown. */
}
```

- A. Write the constructor for the `SumOrSameGame` class. The constructor initializes the instance variable `puzzle` to be a two-dimensional integer array with the number of rows and columns specified by the parameters `numRows` and `numCols`, respectively. Array elements are initialized with random integers between 1 and 9, inclusive, each with an equal chance of being assigned to each element of `puzzle`.

When an element of the two-dimensional array is accessed, the first index is used to specify the row and the second index is used to specify the column.

Complete the `SumOrSameGame` constructor.

```
/**
 * Creates a two-dimensional array and fills it with random integers,
 * as described in part (a)
 * Precondition: numRows > 0; numCols > 0
 */
public SumOrSameGame(int numRows, int numCols)
```

B. Write the `clearPair` method, which takes a valid row index and valid column index as its parameters. The array element specified by those indices, which has a value between 1 and 9, inclusive, is compared to other array elements in `puzzle` in an attempt to pair it with another array element that meets both of the following conditions.

- The row index of the second element is greater than or equal to the parameter `row`.
- The two elements have equal values or have values that sum to 10.

If such an array element is found, both array elements of the pair are cleared (set to 0) and the method returns `true`. If more than one such array element is found, any one of those identified array elements can be used to complete the pair and can be cleared. If no such array element is found, no changes are made to `puzzle` and the method returns `false`.

The following table shows the possible results of several calls to `clearPair`.

puzzle before call to <code>clearPair</code>	Method Call	puzzle after call to <code>clearPair</code>	Return Value	Explanation
0 7 9 0 7 4 1 6 8 4 1 8	<code>clearPair(0, 1)</code>	0 0 9 0 0 4 1 6 8 4 1 8	<code>true</code>	The value 7 in row 0, column 1 is matched with the value 7 in row 1, column 0.
1 2 3 4 5 6 7 8 5 4 1 2	<code>clearPair(2, 2)</code>	1 2 3 4 5 6 7 8 5 4 1 2	<code>false</code>	There is no element in row 2 or a later row to pair with the value 1.
8 1 0 5 0 4 3 6 3 4 5 8	<code>clearPair(1, 1)</code>	8 1 0 5 0 0 3 0 3 4 5 8	<code>true</code>	The value 4 in row 1, column 1 is matched with the value 6 in row 1, column 3. It could also have been matched with the value 4 in row 2, column 1.
1 7 9 2 6 5 4 4 4	<code>clearPair(0, 2)</code>	0 7 0 2 6 5 4 4 4	<code>true</code>	The value 9 in row 0, column 2 is matched with the value 1 in row 0, column 0.

Question 4: 2D Arrays**9 points****Canonical solution**

a. `public SumOrSameGame(int numRows, int numCols)`
`{`
 `puzzle = new int[numRows][numCols];`

 `for (int row = 0; row < numRows; row++)`
 `{`
 `for (int col = 0; col < numCols; col++)`
 `{`
 `puzzle[row][col] = (int) (Math.random() * 9) + 1;`
 `}`
 `}`
`}`

4 points

b. `public boolean clearPair(int row, int col)`
`{`
 `int val1 = puzzle[row][col];`
 `for (int currRow = row; currRow < puzzle.length; currRow++)`
 `{`
 `for (int currCol = 0; currCol < puzzle[0].length; currCol++)`
 `{`
 `int val2 = puzzle[currRow][currCol];`
 `if (currRow != row || currCol != col)`
 `{`
 `if (val1 == val2 || val1 + val2 == 10)`
 `{`
 `puzzle[row][col] = 0;`
 `puzzle[currRow][currCol] = 0;`
 `return true;`
 `}`
 `}`
 `}`
 `}`
 `return false;`
`}`

5 points

a. SumOrSameGame

Scoring Criteria		Decision Rules	
1	Constructs correctly sized 2D array of <code>int</code> and assigns to instance variable <code>puzzle</code>	Responses will not earn the point if they <ul style="list-style-type: none"> fail to update the instance variable <code>puzzle</code>, e.g., by redeclaring as a local variable print or return a value instead of or in addition to updating <code>puzzle</code> 	1 point
2	Traverses 2D array (<i>no bounds errors</i>)	Responses can still earn the point even if they <ul style="list-style-type: none"> fail to construct the 2D array, as long as traversal uses correct bounds fail to assign to the 2D array elements Responses will not earn the point if they <ul style="list-style-type: none"> fail to access an element of the 2D array in the traversal 	1 point
3	Generates a random integer that is uniform in the range <code>[1, 9]</code>	Responses will not earn the point if they <ul style="list-style-type: none"> fail to call <code>Math.random</code> or an equivalent method make any incorrect call to <code>Math.random</code> or an equivalent method call <code>random</code> without <code>Math.</code> fail to cast to an <code>int</code> 	1 point
4	Assigns values to 2D array elements (<i>algorithm</i>)	Responses can still earn the point even if they <ul style="list-style-type: none"> generate the random value incorrectly assign to a local 2D array instead of the instance variable fail to assign some elements due to a bounds error Responses will not earn the point if they <ul style="list-style-type: none"> assign a value other than the attempted randomly generated value correctly initialized the 2D array but now assign somewhere else assign the same value to all visited elements fail to assign to all visited elements 	1 point

b. `clearPair`

Scoring Criteria		Decision Rules	
5	Accesses all and only necessary elements of <code>puzzle</code> in rows <code>>= row</code> (<i>no bounds errors</i>)	<p>Responses can still earn the point even if they</p> <ul style="list-style-type: none"> access additional rows, as long as they also guard against pairing them pair the element at <code>row</code> and <code>col</code> with itself omit extra elements due to incorrect self-pairing guard, as long as the bounds would otherwise support accessing all necessary elements return early, as long as bounds and indices would otherwise support accessing all necessary elements <p>Responses will not earn the point if they</p> <ul style="list-style-type: none"> fail to access elements of <code>puzzle</code> correctly 	1 point
6	Guards against pairing an element with itself	<p>Responses will not earn the point if they</p> <ul style="list-style-type: none"> omit extra potential pairs with their self-pairing guard have no loop 	1 point
7	Determines whether two elements are the same or sum to 10	<p>Responses can still earn the point even if they</p> <ul style="list-style-type: none"> pair the element at <code>row</code> and <code>col</code> with itself have no loop <p>Responses will not earn the point if they</p> <ul style="list-style-type: none"> test only one of the conditions 	1 point
8	Sets a 2D array element to <code>0</code> (<i>in the context of a loop</i>)	<p>Responses can still earn the point even if they</p> <ul style="list-style-type: none"> set only one identified element to <code>0</code> incorrectly identify elements set more than 2 elements to <code>0</code> 	1 point
9	Sets identified elements to <code>0</code> when match is found and returns appropriate <code>boolean</code> in both cases (<i>algorithm</i>)	<p>Responses can still earn the point even if they</p> <ul style="list-style-type: none"> identify elements incorrectly, as long as there is some conditional selection <p>Responses will not earn the point if they</p> <ul style="list-style-type: none"> set only one identified element to <code>0</code> change more than 1 element besides the element at <code>row</code> and <code>col</code> return an incorrect value due to an early return return an incorrect value due to not returning after clearing once 	1 point
Question-specific penalties			
None			