

AP[®] COMPUTER SCIENCE A 2018 SCORING GUIDELINES

Question 4: Latin Squares

Part (a)	<code>getColumn</code>	4 points
-----------------	------------------------	-----------------

Intent: Create a 1-D array that contains the values from one column of a 2-D array

- +1 Constructs a new `int` array of size `arr2D.length`
- +1 Accesses all items in one column of `arr2D` (*no bounds errors*)
- +1 Assigns one element from `arr2D` to the corresponding element in the new array
- +1 **On exit:** The new array has all the elements from the specified column in `arr2D` in the correct order

Part (b)	<code>isLatin</code>	5 points
-----------------	----------------------	-----------------

Intent: Check conditions to determine if a square 2-D array is a Latin square

- +1 Calls `containsDuplicates` referencing a row or column of `square`
- +1 Calls `hasAllValues` referencing two different rows, two different columns, or one row and one column
- +1 Applies `hasAllValues` to all rows or all columns (*no bounds errors*)
- +1 Calls `getColumn` to obtain a valid column from `square`
- +1 Returns `true` if all three Latin square conditions are satisfied, `false` otherwise

Question-Specific Penalties

- 1 (r) incorrect construction of a copy of a row
- 1 (s) syntactically incorrect method call to any of `getColumn()`, `containsDuplicates()`, or `hasAllValues()`

AP[®] COMPUTER SCIENCE A 2018 SCORING GUIDELINES

Question 4: Scoring Notes

Part (a) <code>getColumn</code>			4 points
Points	Rubric Criteria	Responses earn the point if they...	Responses will not earn the point if they...
+1	Constructs a new <code>int</code> array of size <code>arr2D.length</code>		<ul style="list-style-type: none"> only create an <code>ArrayList</code>
+1	Accesses all items in one column of <code>arr2D</code> (<i>no bounds errors</i>)	<ul style="list-style-type: none"> declare the new array of an incorrect size and use that size as the number of loop iterations 	<ul style="list-style-type: none"> switch row and column indices
+1	Assigns one element from <code>arr2D</code> to the corresponding element in the new array		<ul style="list-style-type: none"> use <code>ArrayList</code> methods to add to array
+1	On exit: The new array has all the elements from the specified column in <code>arr2D</code> in the correct order		<ul style="list-style-type: none"> switch row and column indices do not use an index when assigning values to the array
Part (b) <code>isLatin</code>			5 points
Points	Rubric Criteria	Responses earn the point if they...	Responses will not earn the point if they...
+1	Calls <code>containsDuplicates</code> referencing a row or column of <code>square</code>	<ul style="list-style-type: none"> reference any row or column of <code>square</code>, even if the syntax of the reference is incorrect 	
+1	Calls <code>hasAllValues</code> referencing two different rows, two different columns, or one row and one column	<ul style="list-style-type: none"> reference any two distinct rows, two distinct columns, or a row and column of <code>square</code>, even if the syntax of the reference is incorrect 	
+1	Applies <code>hasAllValues</code> to all rows or all columns (<i>no bounds errors</i>)		<ul style="list-style-type: none"> only reference one array in the call to <code>hasAllValues</code>
+1	Calls <code>getColumn</code> to obtain a valid column from <code>square</code>		<ul style="list-style-type: none"> reverse parameters
+1	Returns <code>true</code> if all three Latin square conditions are satisfied, <code>false</code> otherwise	<ul style="list-style-type: none"> test the three sets of conditions and return the correct value 	

Return is not assessed in Part (a).

AP[®] COMPUTER SCIENCE A 2018 SCORING GUIDELINES

Question 4: Latin Squares

Part (a)

```
public static int[] getColumn(int[][] arr2D, int c)
{
    int[] result = new int[arr2D.length];

    for (int r = 0; r < arr2D.length; r++)
    {
        result[r] = arr2D[r][c];
    }
    return result;
}
```

Part (b)

```
public static boolean isLatin(int[][] square)
{
    if (containsDuplicates(square[0]))
    {
        return false;
    }

    for (int r = 1; r < square.length; r++)
    {
        if (!hasAllValues(square[0], square[r]))
        {
            return false;
        }
    }

    for (int c = 0; c < square[0].length; c++)
    {
        if (!hasAllValues(square[0], getColumn(square, c)))
        {
            return false;
        }
    }

    return true;
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.