

4. The `Space` class is used to represent the spaces on a game board. A partial definition of the `Space` class is shown.

```
public class Space
{
    /**
     * Returns the color of the space
     */
    public String getColor()
    { /* implementation not shown */ }

    /**
     * Returns the point value of the space
     */
    public int getPoints()
    { /* implementation not shown */ }

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

The `GameBoard` class maintains a two-dimensional array of `Space` objects that represents a game board. A partial definition of the `GameBoard` class is shown.

```
public class GameBoard
{
    private Space[][] board;

    /**
     * Returns the point value of the row in board specified by the
     * parameter. The point value is the sum of the points in the row,
     * or two times the sum of the points in the row if all spaces in
     * the row are the same color.
     * Preconditions: No elements of board are null.
     *                board has at least two rows and
     *                at least two columns.
     *                targetRow is a valid row index.
     */
    public int getPointsForRow(int targetRow)
    { /* to be implemented */ }

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

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

Write the `GameBoard` method `getPointsForRow`. The method should return the point value of the spaces in the row specified by the parameter `targetRow`. The point value of a row is determined as follows.

- If not all spaces in the row are the same color, the point value for the row is the sum of the points in the row.
- If all spaces in the row are the same color, the point value for the row is two times the sum of the points in the row.

For example, suppose `board` has the following contents. For each element, the first value is the color and the second value is the point value.

	0	1	2	3	4
0	"orange" 100	"red" 100	"blue" 500	"green" 500	"red" 100
1	"red" 300	"blue" 200	"blue" 400	"red" 100	"green" 100
2	"red" 200	"red" 300	"red" 100	"red" 200	"red" 200
3	"green" 100	"blue" 100	"blue" 200	"blue" 100	"blue" 200

For these contents of `board`, the expected behavior of `getPointsForRow` is as follows.

- The call `getPointsForRow(0)` should return 1300. Not all spaces in this row are the same color, so the sum of the points in the row is returned.
- The call `getPointsForRow(2)` should return 2000. The sum of the points in row 2 is 1000, and all spaces in this row are the same color, so two times this sum is returned.

Complete method `getPointsForRow`.

```
/**
 * Returns the point value of the row in board specified by the
 * parameter. The point value is the sum of the points in the row,
 * or two times the sum of the points in the row if all spaces in
 * the row are the same color.
 * Preconditions: No elements of board are null.
 *                board has at least two rows and
 *                at least two columns.
 *                targetRow is a valid row index.
 */
public int getPointsForRow(int targetRow)
```