2. This question involves a scoreboard for a game. The game is played between two teams who alternate turns so that at any given time, one team is active and the other team is inactive. During a turn, a team makes one or more plays. Each play can score one or more points and the team's turn continues, or the play can fail, in which case no points are scored and the team's turn ends. The `Scoreboard` class, which you will write, is used to keep track of the score in a game.

   The `Scoreboard` class contains a constructor and two methods.

   - The constructor has two parameters. The first parameter is a `String` containing the name of team 1, and the second parameter is a `String` containing the name of team 2. The game always begins with team 1 as the active team.
   - The `recordPlay` method has a single nonnegative integer parameter that is equal to the number of points scored on a play or `0` if the play failed. If the play results in one or more points scored, the active team's score is updated and that team remains active. If the value of the parameter is `0`, the active team's turn ends and the inactive team becomes the active team. The `recordPlay` method does not return a value.
   - The `getScore` method has no parameters. The method returns a `String` containing information about the current state of the game. The returned string begins with the score of team 1, followed by a hyphen (`"-"`), followed by the score of team 2, followed by a hyphen, followed by the name of the team that is currently active.

GO ON TO THE NEXT PAGE.

The following table contains a sample code execution sequence and the corresponding results. The code execution sequence appears in a class other than `Scoreboard`.

Statement	Value Returned (blank if none)	Explanation
`String info;`		

`Scoreboard game =`
`   new Scoreboard("Red", "Blue");`
		`game` is a new `Scoreboard` for a game played between team 1, whose name is `"Red"`, and team 2, whose name is `"Blue"`. The active team is set to team 1.

`info = game.getScore();`	`"0-0-Red"`	
`game.recordPlay(1);`		Team 1 earns 1 point because the game always begins with team 1 as the active team.
`info = game.getScore();`	`"1-0-Red"`	
`game.recordPlay(0);`		Team 1's play failed, so team 2 is now active.
`info = game.getScore();`	`"1-0-Blue"`	
`info = game.getScore();`	`"1-0-Blue"`	The score and state of the game are unchanged since the last call to `getScore`.
`game.recordPlay(3);`		Team 2 earns 3 points.
`info = game.getScore();`	`"1-3-Blue"`	
`game.recordPlay(1);`		Team 2 earns 1 point.
`game.recordPlay(0);`		Team 2's play failed, so team 1 is now active.
`info = game.getScore();`	`"1-4-Red"`	
`game.recordPlay(0);`		Team 1's play failed, so team 2 is now active.
`game.recordPlay(4);`		Team 2 earns 4 points.
`game.recordPlay(0);`		Team 2's play failed, so team 1 is now active.
`info = game.getScore();`	`"1-8-Red"`	

`Scoreboard match =`
`   new Scoreboard("Lions", "Tigers");`
		`match` is a new and independent `Scoreboard` object.

`info = match.getScore();`	`"0-0-Lions"`	
`info = game.getScore();`	`"1-8-Red"`	

Write the complete `Scoreboard` class. Your implementation must meet all specifications and conform to the examples shown in the preceding table.

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.