1- Return an array that is "left shifted" by one -- so {6, 2, 5, 3} returns {2, 5, 3, 6}. You may modify and return the given array, or return a new array.

```
public static int[] shiftLeft(int[] nums) {

          int first = nums[0];
          for (int i = 0; i < nums.length - 1; i++) {
              nums[i] = nums[i + 1];
          }
          nums[nums.length - 1] = first;
          return nums;
      }
```

2- Given an array of ints, return the number of times that two 6's are next to each other in the array.

```
public static int array66(int[] nums) {
        int count = 0;

        for (int i = 0; i < nums.length - 1; i++) {
            if (nums[i] == 6 && nums[i + 1] == 6) {
                count++;
            }
        }
        return count;
    }
```

3- In soccer leagues, the winner of a match is awarded with 3 points and the loser 0 points. In case of a tie, both teams are awarded with 1 point each.
Create a class Soccer containing the method maxPoints which takes a int[] **wins**, the number of wins for each team in the league, and a int[] **ties**, the number of ties for each team in the league and returns an int, the maximum points a team in the league has. The *i*'th elements of **wins** and **ties** correspond to the number of wins and ties respectively for team *i*.

Definition

Class:	Soccer
Method:	maxPoints
Parameters:	int[], int[]
Returns:	Int

Method signature: int maxPoints(int[] wins, int[] ties)
(be sure your method is public)

Notes

- Two or more teams may have the same number of points.

Constraints

- **wins** will contain between 1 and 50 elements, inclusive.
- **ties** will contain between 1 and 50 elements, inclusive.
- **wins** will contain the same number of elements as **ties**.
- Each element in **wins** will be between 0 and 100, inclusive.
- Each element in **ties** will be between 0 and 100, inclusive.

```
public int maxPoints(int[] wins, int[] ties) {
	int m = wins[0] * 3 + ties[0];

	for (int i = 0; i < wins.length; i++)
		m = max(m, wins[i] * 3 + ties[i]);
	return m;
}
```