(10/10)

√1. A statistician is studying sequences of numbers obtained by repeatedly tossing a six-sided number cube. On each side of the number cube is a single number in the range of 1 to 6, inclusive, and no number is repeated on the cube. The statistician is particularly interested in runs of numbers. A run occurs when two or more consecutive tosses of the cube produce the same value. For example, in the following sequence of cube tosses, there are runs starting at positions 1, 6, 12, and 14.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Result	1	5	5	4	3	1	2	2	2	2	6	1	3	3	5	5	5	5

The number cube is represented by the following class.

```
public class NumberCube
{
    /** @return an integer value between 1 and 6, inclusive
     */
    public int toss()
    {   /* implementation not shown */   }

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

You will implement a method that collects the results of several tosses of a number cube and another method that calculates the longest run found in a sequence of tosses.

(a) Write the method getCubeTosses that takes a number cube and a number of tosses as parameters. The method should return an array of the values produced by tossing the number cube the given number of times.

Complete method getCubeTosses below.

```
/** Returns an array of the values obtained by tossing a number cube numTosses times.
 *   @param cube a NumberCube
 *   @param numTosses the number of tosses to be recorded
 *           Precondition: numTosses > 0
 *   @return an array of numTosses values
 */
public static int[] getCubeTosses(NumberCube cube, int numTosses) {
    int[] cubeTosses = new int[numTosses];
    for (int i=0; i<numTosses; i++) {
        cubeTosses[i] = cube.toss();
    }
    return cubeTosses;
}
```

Write the method `getLongestRun` that takes as its parameter an array of integer values representing a series of number cube tosses. The method returns the starting index in the array of a run of maximum size. A run is defined as the repeated occurrence of the same value in two or more consecutive positions in the array.

For example, the following array contains two runs of length 4, one starting at index 6 and another starting at index 14. The method may return either of those starting indexes.

If there are no runs of any value, the method returns -1.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Result	1	5	5	4	3	1	2	2	2	2	6	1	3	3	5	5	5	5

Complete method `getLongestRun` below.

```
/** Returns the starting index of a longest run of two or more consecutive repeated values
 *   in the array values.
 *   @param values an array of integer values representing a series of number cube tosses
 *          Precondition: values.length > 0
 *   @return the starting index of a run of maximum size;
 *          -1 if there is no run
 */
public static int getLongestRun(int[] values){
    int currentLen =0;
    int maxLen = 0;
    int maxStart =-1;
    for (int i=0; i< values.length() -1; i++){
        if (values[i] == values[i+1]) {
            currentLen ++;
            if (currentLen > maxLen){
                maxLen = currentLen;
                maxStart = i-currentLen+1;
            }
        }
        else{
            currentLen = 0;
        }
    }
    return maxStart;
}
```