CSCI165 Computer Science II
Week Four Discussion Problems
Array Processing

A statistician is studying sequences of numbers obtained by repeatedly tossing a six-sided die. They are particularly interested in runs of numbers. A run occurs when two or more consecutive tosses of the die produce the same value. For example, in the following sequence of tosses, there are runs starting at positions 1, 5, 12, and 14.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Result	1	**5**	**5**	3	1	**2**	**2**	**2**	**2**	**2**	6	1	**3**	**3**	**5**	**5**	**5**	**5**	1

Problem:

1. Write the static method **int[] tosses(int count).** This method will simulate the tossing of a 6 sided die. The number of tosses will be specified by the **count** parameter. Create an array of integers to store the tosses. Use the **Random** class to simulate the tossing of the die. Return the array when done.

2. Write the static method **int[] runs(int[] tosses)**. This method will accept the array of dice tosses from the method above and create an array that stores all of the start and end points of each run. Using the array above as example, the runs array would store

Tosses Array

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Result	1	**5**	**5**	3	1	**2**	**2**	**2**	**2**	**2**	6	1	**3**	**3**	**5**	**5**	**5**	**5**	1

Runs Array

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Runs	**1**	**2**	**5**	**9**	**12**	**13**	**14**	**17**	0	0	0	0	0	0	0	0	0	0	0

The ***runs*** array should be the same size as the original array but it ***may be*** partially filled.
There may be no runs at all, in which case the runs array will be empty.
There may be runs on each pair, in which case the array will be full.

3. Write the static method **void longestRun(int[] tosses, int[] runs).** This method will determine which run is the longest and will display

   a. Starting and ending indexes for the longest run
   b. The value in the longest run.

4. **Application:** in **main** simulate a test of this process by running it 100 times, 1000 times . . . N times and see if any trends develop. Design a way to have the application display some summary information.

5. Discuss problem solving approaches to this problem and any trends you may notice when running your tests.