ICS4UR—Part I
Programming Assignment 4 – The lights are on but nobody's home
The following program must be programmed in Java
Problem Description
You are walking by a row of K (4 <= K <= 30) lights, some of which are on and some of which are off. In this initial configuration, there may or may not be a consecutive sequence of four lights that are on. Whenever four or more consecutive lights are on, the lights in that consecutive block will turn off. You can only turn on lights that are off.

Write a program that will take a string of zeros and ones in sequence representing lights on (1) and lights off (0). Your program should turn off four or more consecutive lights.

Additional Specifications.

You program should utilize two functions.

Function 1: `public static String findSequenceOfFourInARow(String zAndOnes)`

Function 1 should take in a String as a parameter and return a new string that indicates the beginning and end of a 4-plus consecutive "lights on" sequence.

Example: if zAndOnes was 01111001111011111

The first time you ran it,
Function 1 would return 12 16
Alternately, it could return 1 4

Example: if zAndOnes was 01011100110110100110011

The first time you ran it,
Function 1 would return none

Function2: `public static String turnOffFourInARow(String zAndOnes)`

Function 2 should take in a String as a parameter and return a new string of **0s** and **1s.** The new sequence will be the same except that **one** sequence of 4-plus consecutive "lights on" from the parameter is now off.

Example: if zAndOnes was 01111001111011111

The first time you ran it,
Function 2 would return 01111001111000000
Alternately, it could return 00000001111011111

Input Optional:
Input should be a series of zeros and ones greater than 4 less than 30
If no input is taken you can hard code a number with the same specs. **0s** represent lights off and **1s** represent lights off.

Your program should output a series of strings representing the systematic turning off of lights that are on in a consecutive row of four or more on lights.

Sample Input 1	**Sample Input 2**	**Sample Input 3**
01111001111011111	01101100110110111011	01101100111110111011
Output for Sample Input 1	**Output for Sample Input 2**	**Output for Sample Input 3**
01111001111001111 01111001111000111 01111001111000011 01111001111000001 01111001111000000 01111000111000000 01111000011000000 01111000001000000 01111000000000000 00111000000000000 00011000000000000 00001000000000000 00000000000000000		01101100111110111011 01101100001110111011 01101100000110111011 01101100000010111011 01101100000000111011

Evaluation

Mark	**Level 0 1-49**	**Level 1 50-59**	**Level 2 60-69**	**Level 3 70-79**	**Level 4 80-100**
A3	· Application not complete and does not meet the requirements of the user	· Application complete but meets few of the requirements of the user.	· Application complete, but meets some of the requirements of the user.	· Application complete, and meets most of the requirements of the user.	· Application complete and exceeds the expectations of the user.

.

Based on input tests and code inspection.