**GoofyNumbers Lab Instructions**

**Write out on paper first! (AP style). Then, once approved I will grant you access to the java code and the runner.**

The following Numbers class will be used to analyze and retrieve sets of numbers.

public class Numbers

{

/\*\* @param num is a positive non-decimal value

\* **Precondition :** num >= 0

\* @return true if the sum of digits is odd

@return false if the sum of the digits is even

ex: 2356 🡪 true

2+3+5+6 = 16 🡪 1+6 = 7 🡪 odd

\*/

public static boolean isGoofy(int num)

{

/\* to be implemented in part(a) \*/

}

/\* @param count is a positive non-decimal value

\* **Precondition :** count > 0

\* @return an array containing count Goofy numbers

\*/

public static int[] getSomeGoofyNumbers(int count)

{

/\* to be implemented in part(b) \*/

}

// There may be variables / fields, constructors, and methods that are not shown.

}

A. Write the Numbers method isGoofy(), as started below.  isGoofy() will receive an integer and determine if that integer is goofy or not goofy.

/\*\* @param num is a positive non-decimal value

\* **Precondition :** num >= 0

\* @return true if the sum of digits of num is odd

@return false if the sum of the digits of num is even

ex: 2356 🡪 true

2+3+5+6 = 16 🡪 1+6 = 7 🡪 odd

\*/

public static boolean isGoofy(int num)

{

B. Write the Numbers method getSomeGoofyNumbers(), as started below.  getSomeGoofyNumbers() will receive a count of how many Goofy numbers to return.

The call getSomeGoofyNumbers(3) would return [1, 3, 5].

The call getSomeGoofyNumbers(11) would return

[1, 3, 5, 7, 9, 10, 12, 14, 16, 18, 21].

You must call the method from part a, assuming the method works as specified regardless of what you wrote.  
   
 /\* @param count is a positive non-decimal value

\* **Precondition :** stop > 0

\* @return an array containing count Goofy numbers

\*/

public static int[] getSomeGoofyNumbers(int count)

{