1. What is an infinite loop? Write an example of an infinite while loop.

2. For each of the following loops indicate how many times its body will execute.
   
   (a) for(int i = 1; i <= 10; i++)
   (b) for(int i = 1; i < 10; i++)
   (c) for(int i = 1; i < 10; i--)
   (d) for(int i = 1; i > 10; i++)
   (e) for(int i = -19; i < 0; i+=4)

3. Write a loop over an integer that counts from \texttt{START} to \texttt{END} inclusive and, at each step, calls the method \texttt{cuckooForCocoaPuffs(int)} on twice the value of the loop control variable. Do this for each of the following three loop constructs.
   
   (a) for

   (b) while

   (c) do while
(d) Can the preceding loops ever produce different results for the same values of START and END? If not, why? If so, when?

4. Write a simple program that prompts a user to enter a list of integers. The user should be prompted for each integer and signifies the end of the list with an X or an x. When the user is finished, print out the average value of the entire list. (Hint: use `nextLine()` to get input from the user, and `Integer.parseInt()` to convert to integers)

5. Provide the implementation for the `hokeyGraphics(char cell, int size, boolean isRect)` method, which prints out a shape made of the cell character and having the given dimensions. If `isRect` is `true` print out a rectangle, otherwise print out a triangle. Here are some examples.

- `hokeyGraphics('*', 3, true)` produces
  
  ***
  ***
  ***

- and `hokeyGraphics('@', 3, false)` produces
  
  @
  @@
  @@@

6. Write a method called `divisors(int x)` that takes a positive integer and prints out all of its divisors (except 1). For example `divisors(36)` should output something like `Divisors of 36 are: 2, 3, 4, 6, 9, 12, 18`. Bonus: write a method that prints out the prime factorization of x (i.e. only those divisors of x that are prime and, for each such divisor, the number of times it goes in to x. `primeFactors(36)` would output something like `Prime factorization of 36 is: 2*2*3*3`).