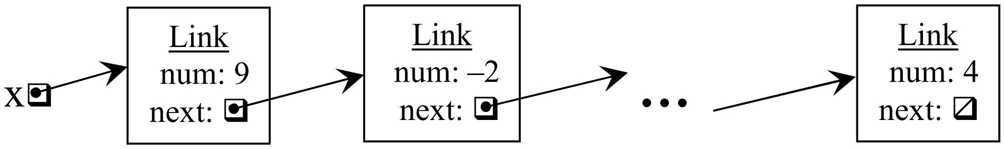
Linked List Exercises, CSC 143

The variable 'x' is bound to a linked list of **unknown length**. For parts 1 and 2 below, write a few lines of code (not method definitions) to solve the problems without changing the value of 'x' itself. 'num' and 'next' are public, and 'num' is type *int*. Parts 3 & 4 are separate problems using the same Link class.

HINT: Draw pictures to figure out how to solve the problem!



Warm-Up: What does this code do?

```
boolean found = false;
Link p0 = x;
while ( !found && p0 != null ) {
  if ( p0.num == 8 )
    found = true;
  p0 = p0.next;
}
System.out.println( found );
```

1. Write code to determine the **average** of all the numbers on the list without changing the list.

2. Write code that declares a variable 'lowest' and makes it **"point to"** (act as a reference to) the Link with **the lowest number**. Be careful about initialization and special cases.

3. Implement this method:

```
  /**
   * 3. Make a new list from the numbers in the int[] nums.
   * Return null if nums is null or empty.
   * @param nums the source array
   * @return a Link to the first element in the list
   */
  public static Link makeList( int[] nums ) {
















}
```

4. Implement this method:

```
  /**
   * 4. Create a String representing the list using
   * this format: {7,5,-2,8}
   * @param head the source list
   * @return a String representation
   */
  public static String listToString( Link head ) {
















}
```