- Declare a variable to hold the precise weight of a sheet of paper in pounds (or kilograms).
- Write a snippet of code that determines of a variable called **weight** is bigger than 0 but less than 1. If it is, output lightweight to the console.
- Write a loop that allows the user to enter (via the console) a **target** weight. This is different than the **weight** variable above. Force the user to re-enter the value if they give a target weight below 0 or above 1.
- Write a function that computes the number of sheets of paper that can be contained in the target weight. (This is a simple division.) We need the **target** weight of the package and the **weight** of a single sheet of paper to do this computation
- Create a vector of target weights and allow the user to input as many weights as the user desires. Use negative values to indicate that the user no longer wants to enter any values.
- Use your function to compute the number of sheets of paper that can be contained in each of the target weights in the vector. Place the result in a new vector.
- Write a function to output (to the console) every element of a vector and use it to output the last vector you've created.
- Create a class called **Paper** that has an internal weight variable. Write setters and getters for the weight. Use good programming practice.
- Read in 10 weights from a file called "apples.txt" and store them in a vector of **Paper**.
- Do the same as the previous bullet, but for each weight, create a dynamic **Paper** object using new. Set its weight and then put them in a vector. Specifically, you'll be storing them in a vector of **Paper***.
- Recursively compute the weight of a sheet of paper after tearing it in half *n* number of times. It follows the following functionality:

  $paper^0$ = paper's weight --- 0 rips

  $paper^n = paper^{n-1} / 2$ --- after ripping in half n times

  - Extra points for figuring out what this function is actually doing!