

## Homework 4: Leonardo Numbers

Your assignment is to write a MIPS program that inputs a number  $n$  from the user, and then outputs the 0th to  $n$ th Leonardo numbers\*. The final program you turn in should be called `leonardo.asm`.

Leonardo numbers are similar to Fibonacci numbers, and are defined recursively as follows:

$$\begin{aligned}L_0 &= 1 \\L_1 &= 1 \\L_i &= L_{i-1} + L_{i-2} + 1\end{aligned}$$

You must print every Leonardo number up to the element given by the user. Here are some examples:

```
Please enter a nonnegative integer: 0
1
```

```
Please enter a nonnegative integer: 1
1, 1
```

```
Please enter a nonnegative integer: 5
1, 1, 3, 5, 9, 15
```

```
Please enter a nonnegative integer: -1
Try again.
```

This problem is solvable with a simple loop; you do not need to use recursion or an array. (Though you are welcome to, if you desire. However recursion is difficult and inefficient, and an array should be properly allocated to hold exactly  $(n + 1)$  elements.) Sketch out how you'd solve this with a simple loop in Java or C, and then implement it in assembly.

Coding style is different than in a high-level language, since assembly is so unreadable. It is expected that most lines of your code will have comments and that you will do your best to make your code comprehensible.

---

\*Leonardo numbers are primarily used by a sorting algorithm called smoothsort, written by Edsger Dijkstra in 1981. It is a variation of heapsort, but its time complexity approaches  $O(n)$  on mostly-sorted data.