

CS119 – Lab 4
Due Date: March 6

Purpose: Abstracting common patterns on computation into a function is a powerful tool in computer science, since it makes it so we don't have to duplicate a lot of work over and over again. We have seen three such abstractions in the functions `every`, `keep`, and `accumulate` and we will be using these three functions in this lab to illustrate how they can be used to solve many different problems.

Knowledge: This lab will help you become familiar with the following content knowledge:

- How to use the higher order functions `every`, `keep`, and `accumulate`

Task: Follow the steps in this lab carefully to complete the assignments. Copy the lab4 folder and complete the following assignments by writing functions in `Example4.hs`. In all of these assignments, you will not have to use recursion since all the recursion is built into the higher order functions.

Assignment 1:

Use `every` to write a function `exaggerate` that takes a sentence and doubles all the numbers in the sentence and replaces the word "good" with the word "great and the word "bad" with the word "terrible":

```
> exaggerate (sent "I ate 3 good hotdogs")  
[I ate 6 great hotdogs]
```

Hint: You need to write the function that will be applied to each word in the sentence. This function checks for the special cases and returns the appropriate results. Otherwise it should just return the word unchanged.

Criteria for Success: You have written a function that when passed as a parameter into `every` gives you the body of your function `exaggerate`, behaving like the given example.

Assignment 2:

Use `keep` to write a function `firstLast` that keeps only the words in a sentence whose first and last letters are the same:

```
> firstLast (sent "california ohio nebraska alabama maryland")  
[ohio alabama]
```

Criteria for Success: You have written a function that when passed as a parameter into `keep` gives you the body of your function `firstLast`, behaving like the given example.

Assignment 3:

Use `accumulate` to write a function `hyphenate` that hyphenates all the words of a sentence together.

```
> hyphenate (sent "one thousand forty five")
one-thousand-forty-five
```

Hint: The combiner function used in `accumulate` must take two words and combine them to return a single word.

Criteria for Success: You have written a function that when passed as a parameter into `accumulate` gives you the body of your function `hyphenate`, behaving like the given example.

Assignment 4:

Use `every`, `keep` and `accumulate` to write a function `acronym`.

```
> acronym (sent "reduced instruction set computer"
risc

> acronym (sent "foundations of computer science")
fcs
```

Small connecting words like "of" are not part of the acronym. You may use the function `realWord` to determine if a word is irrelevant or not.

Criteria for Success: You have used a combination of the higher-order functions to define `acronym`, behaving like the given examples.

Submit your `Example4.hs` file in Canvas for grading.