CS3 Midterm Review

More complicated recursion

1. Define a function uncompress which given a sentence of alternating numbers and words returns a sentence with each word duplicated by the number preceding it.

   ```
   (uncompress ‘(3 ha 1 you 1 are 4 so 1 weird))
   → (ha ha ha you are so so so so weird)
   ```

2. Produce a function to find the length of the longest decreasing substring in a sentence of positive numbers. For example:
   (2 1 5 3 2 1 12)→ 4
   (13 12 11 15 10 5 2 1) → 5

3. Write a procedure that returns all words representing subsets of the letters in the input sentence. Assume that the input sentence is not empty and includes distinct letters. Note that the order does not matter in out put subsets so `ab` is considered the same subset as `ba`. Example:
   (a b c) → ("", a, b, c, ab, ac, bc, abc)
   (w e) → ("", w, e, we)

4. I have c chips and d drinks, how many ways can I finish all these snack if I ate one at a time? Example: `(snack 1 2) → 3`, which includes (chip, drink, drink), (drink, chip, drink), and (drink, drink, chip).

Higher-Order-Functions

5. Write a function add-digit which will attach a number to the end of each word in the sentence.

   ```
   (add-digit 3 ‘(hi how are you doing)) → (hi3 how3 are3 you3 doing3)
   ```

6. Write function that finds the mode (or one of the modes in case of ties) of the numbers in the input sentence. Example:
   (1 2 3 12 1 1)→ 1
   (12 3 5 3 12) → 12 or 3

7. Consider a database of student information that consists of a word for each student of with the following structure: `Name,Age,Letter-Grade`. Assuming that student name includes only English letters, write a procedure that returns a sentence of all student ages. For instance:
   `(Mary,19,A+ Bill,23,B- Jimmy,44,A Sally,25,C+)` → `(19 23 44 25)`

8. Modify your procedure so that it returns the name of (one of) the oldest student(s) in class. So for the previous example the procedure should return: `Jimmy`.

9. Earlier in lab you practiced writing the procedure `down-up` in recursion. Now let's do it using HOF! `(downup ‘clint)` → `(clint clin cli cl c cl cli clin clint)`

Lambda

10. What does the following evaluate to? If it returns an error, describe what the error is and why it occurred. If it returns a procedure, describe what argument(s) it takes and what it does with it/them. If it returns a value, write out the arithmetic expression of it.

```
STk> ((lambda (x y)
         (lambda (a b)
             (+ a ((lambda (g h) (* g h)) b 10) x y))) 5 8)
```

11. If the previous expression generated an error, fix it and make a procedure call. If it returned a procedure, how do you make a procedure call with arguments 10 and 7 without changing the existing code, and what does it return? (i.e. don't add anything inside the left-most parenthesis and the right-most parenthesis.)

12. Write a function that categorizes all names in a sentence based on their first letter. It should put all names starting with A first; then all names starting with B; and so on… For instance if the input sentence is `(Adam Bobby John Catherine Sara Bill Abraham Jim)`, the function will return `(Adam Abraham Booby Bill Catherine John Jim Sara)`. [this is the first step to implement bucket sort!]