Project 4 – Scheme

Complete each of the following Scheme procedures using basic Scheme primitives. If you need a helper function you should implement it yourself. All procedures should be written in functional style. You can rely on the return value for output – do not display output in your procedures.

Helper functions are your friend.

Your submission should be a *single* .scm or .rkt file with the code below appended at the end. Your code should compile. If you do not complete any of the procedures then comment out the call to that procedure in the test code. Otherwise, you should make *no changes* to the test code or the proj4test.scm module. Your code should import proj4test.scm, but do not copy that code to your project file. Your program will be tested with a different proj4test.scm that follows the same structure.

1. Define a *recursive* Scheme procedure `(removes-last-negative lst)` which returns the last negative number removed. (4 points)

   For example: `(return-last-negative '(1 -2 -3 4))`

   ⇨ `(1 -2 4)`

2. Write a Scheme procedure `(distance a b)` which, given two pairs representing points on the Cartesian coordinate system, returns the distance between the two points.

3. Write a Scheme procedure `remove-dups` which takes a list of words/letters as input and returns the list with consecutive repeated words/letters removed. Note that multiple instances of a letter are allowed as long as they are not consecutive.

   For example, `(remove-dups ‘(a b c c d d c a a b)) -> ‘(a b c d c a b)`

4. Write a Scheme procedure called `(squares-up-to x)` that returns a list of squares less than or equal to the parameter.

   For example: `(squares-up-to 30) -> (1 4.0 9.0 16.0 25.0)`

5. Write a Scheme procedure called `(election-results lst)` that calculates the winner of a US presidential election. Recall that the president is chosen not by popular vote but by the Electoral College. The candidate who receives the most votes in a state receives all of that state’s electoral votes. The input `lst` is a state-by-state list of pairs, one for each state. The car of the par is the number of electoral votes that state has and the cdr is a pair of results for two candigtates (assume there are only two candidates (A & B) and that the results for A are always listed first). For example, in the following case candidate A wins the electoral votes in all four states and therefore has the most number of total electoral votes:

```
(define sample '(
   (9 . (941173 . 692611))
   (3 . (167398 . 79004))
   (8 . (781652 . 685341))
   (6 . (472940 . 422768))))
(election-results sample)
 -> A
```

Include the code below at the end of your program.

```
; Test code
(test-remove-last-negative remove-last-negative)
(test-distance distance)
(test-remove-dups remove-dups)
(test-squares-up-to squares-up-to)
(test-election-results election-results)
```