CS117 (Fall 2019) Programming Assignment 6 20 pts. Due: Dec. 05 (Th)

Practice for functional programming with Scheme

Task: Implement the merge-sort in Scheme.

To implement the merge-sort in scheme you need to define three functions, i.e., split, merge, and mergesort.
Create a file (e.g., "merge-sort") in your working directory including the following three functions:

```
(define (split x)
  (cond . . . .
   . . . .
   )
 )

(define (merge x  y)
  (cond . . . .
   . . . .
   )
 )

(define (mergesort x)
  (cond . . . .
   . . . .
   )
 )
```

Save this file and invoke the scheme interpreter that you installed in your computer.
A sample run time session is shown in the next page.

Include good documentations in your code and submit the hardcopy of your source code and run time output.

A sample run time session

```
1 ]=> (load "merge-sort")

;Loading "merge-sort"... done
;Value: mergesort

1 ]=> (split '())

;Value 13: (())

1 ]=> (split '(1))

;Value 14: ((1))

1 ]=> (split '(1 2))

;Value 15: ((1) 2)

1 ]=> (split '(1 2 3 4))

;Value 16: ((1 3) 2 4)

1 ]=> (merge '(1 2 3 4) '(5 6 7 8))

;Value 17: (1 2 3 4 5 6 7 8)

1 ]=> (merge '(4 3 2 1) '(8 7 6 5))

;Value 18: (4 3 2 1 8 7 6 5)

1 ]=> (merge '(2) '(1))

;Value 19: (1 2)

1 ]=> (merge '(1 3) '(2 4))

;Value 20: (1 2 3 4)

1 ]=> (mergesort '(7 8 5 6 4 3 2 10))

;Value 21: (2 3 4 5 6 7 8 10)

1 ]=> ^D
```