```
CPSC/ECE 3220 - Fall 2017 - Project 2 Assignment

  Due date:    Thursday, October 12, 2017 by 11:59PM
  Submission:  turn in to Canvas a copy of your source code and write up .pdf
  Grading:     50% Code and Correctness
               20% Comments Either block/section comments or line-by-line comments
are acceptable.
               30% a 1-page write up demonstrating difficulties of threaded program
and debugging techniques for proving correctness.

  This can be done as an individual assignment or as a team of two.

  Assignment: Page 179; Exercise 6
              Write a program that uses threads to perform a parallel merge sort.

  Testing:    ./program  <input.txt  >output.txt

  Notes:      A program that does NOT use treads will receive a 0.
              Use scanf() / printf() for input / output or other stdin / stdout
system calls.
              Threads run out-of-order and will to use system class to keep
synchronization.
              input.txt will be a list of 0 to 100000 numbers inclusive.
              Check for zero inputs and one input as special conditions.
              Use may use an array[100000] as storage for the inputted numbers.

              input.txt
              ==========================
              5
              6
              3
              2
              ==========================

              output.txt will be a list of running commentary of the program followed
by a list of sorted numbers.
              Remember threads execute out of order.
              ==========================
              ...
              Creating thread for left half of list index 0-1
              Creating thread for right half of list index 1-1
              Creating thread for left half of list index 0-0
              Creating thread for right half of list index 2-3
              Creating thread for left half of list index 2-2
              Exiting thread for index 1-1
              Exiting thread for index 0-0
              Merging for index 0-1
              Exiting thread for index 0-1
              ...
              ...
              The List Sorted:
              2
              3
              5
              6
              ==========================
```