**Programming Assignment 2**

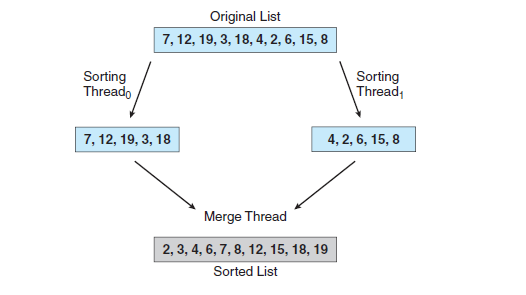
# Objective

The objective of this assignment is to understand the use of threads and how threads can be used in multithreaded programs.

# Assignment: Multithreaded Sorting Application

Write a multithreaded sorting program that works as follows: A list of integers is divided into two smaller lists of equal size. Two separate threads (which we will term sorting threads) sort each sub list using a sorting algorithm of your choice. The two sub lists are then merged by a third thread—a merging thread —which merges the two sub lists into a single sorted list.

Since global data are shared cross all threads, perhaps the easiest way to set up the data is to create a global array. Each sorting thread will work on one half of this array. A second global array of the same size as the unsorted integer array will also be established. The merging thread will then merge the two sub lists into this second array. Graphically, this program is structured as shown below:



Name the program *thrsort.c* or *thrsort.cpp.* You can use the Original List depicted in the figure above for testing but assume the size and values of the list can change. Example run of *thrsort* is shown below:

* *./thrsort*
* *Sorted List: 2, 3, 4, 6, 7, 8, 12, 15, 18, 19*

# Grading

The program will be graded on the basic functionality, error handling and how well the implementation description was followed. Be sure to name your program ***thrsort.c (thrsort.cpp)*** (no extra characters, capitals) Note that documentation and style are worth 10% of the assignment's grade!