* **Code Reflection:** A brief explanation of the code and its purpose, and a brief discussion of your experience in developing it, including any issues that you encountered while completing the exercise and what approaches you took to solve them.
  + This code sorts (alphabetically) a list. Any csv list can be read and sorted. The test lists used were from an eBid site. It sorted all the items sold. It was a little confusing to develop at first, I wasn’t exactly sure what I was doing with it, but after looking over the code we were given, I was able to add the sorting algorithm. While testing, I was having an issue with it sorting all the way with the QuickSort. About 1 in 10 items weren’t sorted correctly.

**Pseudocode or Flowchart:** A pseudocode or flowchart description of the code that is clear and understandable and captures accurate logic to translate to the programming language.

START

RUN main()

DISPLAY menu

GET user input

IF “1”:

Calls loadBids(): Loads values from provided .csv file

Stores them as vector “bids”

Default: “eBid\_Monthly\_Sales\_Dec\_2016.csv”

IF “2”:

Displays all values from vector “bids”

FOR EACH bid, PRINT)

IF “3”:

START clock

Calls selectionSort(): Sort vector “bids” using Selection Sort

FOR EACH position in the list, set that number as “min”

(FOR loop) loop through everything to the right of the “min”

IF this element's title < “min” title

The element is set to the “min”

IF the “min” isn’t the position

Swap them

END clock

Print: the amount of time taken in seconds.

IF “4”:

START clock

Calls quicksort(): Sort vector “bids” using Quick Sort

INITIALIZE “mid” to 0

Set mid = Calls partition()

Pivot = the middle of the list: (low + (high-low)/2)

SET done = FALSE

While not done

While the low bid < pivot

Increment low

While the pivot < high bid

Decrement high

IF low bid >= high

Done = TRUE

ELSE

Swap low bid and high bid

Increment low

Decrement high

Return high bid

END clock

Print: Amount of time taken in seconds.