**Project 2 Code Reflection**

**Tammy Hartline**

**03/11/2023**

**CS-300**

I found this assignment to be a bit of a challenge. Albeit mainly because I was overcomplicating what was already given. I tend to have a harder time with assignments when things are already created because I feel the need to look through everything and understand its purpose before I begin any edits. Once I felt I had that understanding, the rest went smoothly. That is, until I went to run it and it could not find the excel files. This was due to me loading it incorrectly. After reloading it SEVERAL times and moving things around, and finally walking away from it until the next day, I was able to get it to load and everything executed successfully and as intended.

The purpose of my code was to implement partitions, selection sorts, and quick sorts, in order to get a clearer view of items sold in an eBay auction. Selection and quick sort was done based on the auction title. To get the program to execute correctly and as intended, I needed to ensure that I had an understanding of the partition and pivot point functionality. Once that was understood, I did not have any coding difficulties, just user error ones, lol.

I am not a huge fan of flowcharts as I tend to over document my code with comments, lol. I am getting better about not cluttering up my code after receiving some feedback from Peer QA with my last development submitted at work. So I am choosing to do pseudocode for my documentation, but still included ample comments in my project as well and tried to not over complicate my pseudocode either, lol.

**Pseudocode For Project 2 CS-300**

2a – Quick Sort (Partition)

DECLARE INT as low and set equal to begin

DECLARE INT as high and set equal to end

DECLARE INT pivot and SET as equal to low plus (high minus low) divided by 2 (Calculation)

DECLARE BOOL as is done and set equal to false

WHILE is done IS NOT true

AND

WHILE bids title is low when compared to pivot IS less than 0

THEN Increment low

WHILE bids title is high when compared to pivot IS less than 0

THEN decrement high

IF low is greater than or equal to high

Is done is set to TRUE

ELSE

SWAP low with high

Increment low

Decrement high

RETURN high

GO TO VOID quick sort

DECLARE INT mid point and set equal to 0

IF begin is greater than or equal to end

RETURN

SET mid point equal to partition(bids, begin, end)

CALL quick sort(bids, begin, mid point)

CALL quick sort(bids, mid point plus 1, end)

GO TO selection sort

DECLARE INT min

DECLARE INT size\_t and set equal to bids size

FOR UNSIGNED pos is equal to 0 AND pos is less than size\_t THEN increment pos

Set min equal to pos

FOR UNSIGNED j is equal to pos plus 1 AND j is less than size\_t, increment j

DECLARE STRING current bid title and set equal to bids at j title

DECLARE STRING minimum bid title and set equal to bids at min title

IF min IS NOT equal to pos

SWAP bids at pos with bids at min

ADD CASE 3 WHERE

INITIALIZE ticks and set equal to clock

CALL selection sort

CALCULATE time elapsed with ticks is equal to clock minus ticks

OUTPUT calculations

ADD CASE 4 WHERE

INITIALIZE ticks and set equal to clock

CALL quick sort

CALCULATE time elapsed with ticks is equal to clock minus ticks

CALCULATE time elapsed with ticks is equal to clock minus ticks

BREAK