**Reflection**

The program is designed to import data from a CSV file, parse that file, and store the data in a linked list as a collection of “Bid” objects. From there, the program enables a user to add or remove bids, as well as search for specific bids in the list. The prepend and append functions allow for inserting bids either at the beginning or the end of the list, while the search and remove functions traverse the list to either display the data or delete it. The biggest issues encountered in this assignment were list manipulation, ensuring that the code referenced the correct node, whether it was a head or a tail, and ensuring the functions traversed properly. Another significant challenge was enabling user input for the file path and implementing the necessary error handling. In terms of performance, the list was designed with head and tail pointers, providing O(1) time complexity for appending or prepending to the list. Additionally, the CSV file operations were enclosed in try-catch blocks to ensure graceful error handling.

**Pseudocode**

Function Prepend(bid):

Create new node with bid

new node.next = head

head = new node

If tail is null:

tail = head // List was empty

Increment size counter

Function PrintList():

Set current node = head

While current node is not null:

Print current node’s bid (bidId, title, amount, fund)

Move to next node (current = current.next)

Function Remove(bidId):

Set current = head

Set previous = null

While current is not null:

If current.bidId matches bidId:

If current is head:

head = head.next

If current is also tail:

tail = null

Else:

previous.next = current.next

If current is tail:

tail = previous

Delete current node

Decrement size counter

Return

previous = current

current = current.next

Function Search(bidId):

Set current = head

While current is not null:

If current.bidId matches bidId:

Return current.bid

Move to next node (current = current.next)

Return empty bid (not found)