**4-2 Assignment: Hash Tables**

**Reflection:**

In this project I worked on building a hash table to organize and manage bids from the CSV file. The program lets you load bids, search for one by its ID, remove a bid, and display all of them. I used a hash function to turn the bid ID into a key and I used a linked list to handle situations where two bids had the same key. One challenge I had was setting up the program in Visual Studio. At first I got errors saying the program couldn’t start, and later the CSV wouldn't load right. I fixed it by choosing the correct startup file and checking my project settings. Another challenge was making sure the Insert, Search, and Remove functions worked correctly when multiple bids shared the same hash key. I used chaining to handle those situations. I learned a lot about how hash tables work in real programs, especially how to handle collisions and search through linked nodes. Now I feel more confident working with data structures like this in future projects.

**Pseudocode:**

**Main Menu Logic**

If choice equals 1

 Start timer

 Call loadBids() with CSV path and hash table

 Output number of bids loaded

 Stop timer

 Output time in clock ticks and seconds

End If

If choice equals 2

 Call PrintAll()

End If

If choice equals 3

 Ask for bidId input

 Start timer

 Call Search(bidId)

 If bid is found

  Output bid info

 Else

  Output "Bid Id not found"

 Stop timer

 Output time

End If

If choice equals 4

 Ask for bidId input

 Call Remove(bidId)

End If

If choice equals 9

 Output "Good bye"

 Exit program

End If

**HashTable::Insert(bid)**

Call hash() using bid.bidId → store in key

If nodes[key] is null

 Create new node with bid

 Assign to nodes[key]

Else if nodes[key] is empty (UINT\_MAX)

 Set bid and key in node

Else

 Set current = nodes[key]

 While current->next not null

  Move to next

 Create new node with bid

 Attach to end of list

End If

**HashTable::PrintAll()**

For i from 0 to tableSize

 Set current = nodes[i]

 While current is not null

  Output bidId, title, amount, fund

  Move to current->next

End For

**HashTable::Search(bidId)**

Call hash(bidId) → store in key

Set current = nodes[key]

While current is not null

 If current->bid.bidId equals bidId

  Return bid

 Move to next node

Return empty bid

**HashTable::Remove(bidId)**

Call hash(bidId) → store in key

Set current = nodes[key]

If current is null

 Return

If current->bid.bidId equals bidId

 If current->next is null

  Delete node

  Set to null

 Else

  Copy next node data into current

  Delete next node

 Return

Else

 While current->next is not null

  If current->next->bid.bidId equals bidId

   Set temp = current->next

   Set current->next = temp->next

   Delete temp

   Return

  Move to next node

End If