CSE220: Data Structures (Lab)
Fall 2024
Lab Quiz - 04
Duration: 30 Minutes

B

Name:	ID:	Section:

Question 1 [15 Points]

In this task, you are asked to implement a **HashTable** class that stores key-value pairs, where the key is **a string (representing a product ID)** and the value **is a float (representing the product's price)**. The class should include a **hash_function** that computes the hash index based on the sum of the ASCII values of the first three characters of the key. If the key is shorter than three characters, it should add the ASCII value of '0' (48) to make the key length three.The **insert()** method should insert a new key-value pair into the hash table. If a collision occurs, the method will use forward chaining (linked lists) to store multiple entries at the same index. If the key already exists, it should update the value by adding the new price to the previous price.

You are not allowed to use any built-in functions except len(). Assume the display method is already implemented.

Sample Input:	Sample Output:	Explanation:

ht = HashTable(10)
 	
ht.insert("P123", 19.99)
ht.insert("AB", 15.50)
ht.insert("P456", 25.75)
 	
print("\nHash table after insertions:")
ht.display()
 	
ht.insert("P123", 21.99) # Updating price for P123 by adding the new price
 	
print("\nHash table after update:")
ht.display()
	Hash table after insertions:
	Index 5: P456 (25.75)
	
	Index 9: P123 (19.99)
	A45 (15.50)
	
	Hash table after updates:
	Index 5: P456 (25.75)
	Index 9: P123 (41.98) # 19.99 + 21.99
	Index 9: A45 (15.50)
		For P123,Hash function calculation, 'P' = 80, '1' = 49, '2' = 50.Total sum = 80 + 49 + 50 = 179. So, index=179%10=9
		For AB ‘A' = 65, 'B' = 66. Since the key is less than 3 characters, the ASCII value of '0' (48) is added to the sum.Total sum = 65 + 66 + 48 = 179
		So, index = 179 % 10 = 9
		When we try to insert P123 again, since the key already exists its value will be updated by adding 19.99 with 21.99.