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

A

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 an employee ID)** and the value is a **string (representing the department name)**. The class should include a **hash_function** that computes the hash index based on the sum of ASCII values of each character in the key, then Multiply each ASCII value by its position index (starting from 1) and finally sum these weighted values and take the modulus with the size of the hash table. The **`insert()`** method should insert a new key-value pair or update the value if the key already exists, using **forward chaining** to handle collisions. **If the key already exists, its value should be updated. [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(4)`
 	
`ht.insert("E123", "HR")`
`ht.insert("BA", "Finance")`
`ht.insert("XY", "Engineering")`
`ht.insert("YX", "Marketing")`
 	
`print("\nHash table after insertions with collisions:")`
`ht.display()`
 	
`ht.insert("E123", "Admin")`
`print("\nHash table after update:")`
`ht.display()`
	Hash table after insertions with collisions:
	Index 0: (BA: Finance) -> None
	Index 1: (E123: HR) -> (YX: Marketing) -> None
	Index 2: (XY: Engineering) -> None
	Index 3: None
	
	Hash table after update:
	Index 0: (BA: Finance) -> None
	Index 1: (E123: Admin) -> (YX: Marketing) -> None
	Index 2: (XY: Engineering) -> None
	Index 3: None
		For E123:
		
		ASCII values: 'E' = 69, '1' = 49, '2' = 50, '3' = 51.
		Weighted sum: (1 × 69) + (2 × 49) + (3 × 50) + (4 × 51) = 69 + 98 + 150 + 204 = 521.
		Index: 521 % 4 = 1.
		
		For BA:
		
		ASCII values: 'B' = 66,'A' = 65.
		Weighted sum: (1x66)+(2 × 65) = 296.
		Index: 296 % 4 = 0.