

Name:	ID:	Section:
-------	-----	----------

### Question 1 [15 Points]

You are tasked with implementing a **hash table** that stores student records. Each record contains a **student name (a string)** and a **student ID (an integer)**. You will implement a hash table that uses **forward chaining** (a linked list) to handle collisions. Implement the following methods:

**Hash Function:** You are given a string representing the student name. You need to calculate a hash index for this string using the following rules: Take the sum of the ASCII values of the characters in the string. If the sum is odd, return the sum modulo the size of the hash table. If the sum is even, return the sum divided by 2, modulo the size of the hash table.

**Insert:** Implement the `insert()` method that takes a student name (string) and student ID (integer) and stores them in the hash table. If a student in the same index already exists in the hash table (i.e., there is a collision), use forward chaining (linked list) to store multiple student records at the same hash index. **Note:** If the key-value pair already exists, you will print an error message saying “Student already exists” and discard the key-value pair.

**Search:** Implement the `search()` method that takes a student name (string) and returns the corresponding student ID (integer). If not found, then return `None`. **[No built-in function except `len()`. Assume the display method and Node class are already implemented]**

Sample Input:	Sample Output:	Explanation:
---------------	----------------	--------------

<pre>ht = HashTable(5) ht.insert("Alice", 123456) ht.insert("Bob", 135927) ht.insert("Charlie", 348247) ht.insert("David", 124382) print("\nHash table after insertions and rehashing:") ht.display()  ht.insert("Alice", 123456) ht.search("David")</pre>	<pre>Hash      table      after insertions: Index    0:      ("Bob", 135927) Index 1: None Index 2: None Index    3: ("Charlie", 348247) Index    4:      ("Alice", 123456)-&gt;("David", 124382)  Student already exists  ID of David is 124382</pre>	<p>For 'Alice', hash function calculation, sum of ASCII values = 65 + 108 + 105 + 99 + 101 = 478. Since 478 is even, the hash index = <math>478 / 2 \% 5 = 4</math>.</p> <p>For 'Bob', sum of ASCII values = 66 + 111 + 98 = 275. Since 275 is odd, the hash index = <math>275 \% 5 = 0</math>.</p> <p>When we try to insert ("Alice", 123456) again, since the (key, value) pair already exists, an error message is printed.</p> <p>David is found at index 4 and his ID returned.</p>
--	--	--