Name:	ID:	Section:

You are asked to implement a HashTable class that stores key-value pairs, where the key is a string (representing a student ID) and the value is an integer (representing the student's grade).

The class should include a hash_function that calculates the hash index by summing the ASCII values of the **first two characters** of the key. If the key is only one character, the ASCII value of 'Y' (89) should be added as the second character.

The insert() method should add a new key-value pair or update the value if the key already exists. Use **forward chaining** to handle collisions.

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

Python

Sample Input	Sample Output	Explanation
```# Sample Input ht = HashTable(7) ht.insert("S1", 85) ht.insert("A", 90) ht.insert("S2", 78) print("\nHash table after insertions:") ht.display() ht.insert("S1", 88) # Updating S1 grade print("\nHash table after update:") ht.display()```	```Hash table after insertions: Index 0: A (90) -> S2 (78) -> None Index 1: None Index 2: None Index 3: None Index 4: None Index 5: None Index 6: S1 (85) -> None Hash table after update: Index 0: A (90) -> S2 (78) -> None Index 1: None Index 2: None Index 3: None Index 4: None Index 5: None Index 6: S1 (88) -> None```	**For S1**: 'S' = 83, '1' = 49. Total = 83 + 49 = 132. Index = 132 % 7 = 6. **For A**: 'A' = 65, 'Y' = 89. Total = 65 + 89 = 154. Index = 154 % 7 = 0. **For S2**: 'S' = 83, '2' = 50. Total = 83 + 50 = 133. Index = 133 % 7 = 0.

JAVA

Sample Input	Sample Output	Explanation

``
// Sample Input
HashTable ht = new HashTable(7);
ht.insert("S1", 85);
ht.insert("A", 90);
ht.insert("S2", 78);
System.out.println("\nHash table after insertions:");
ht.display();
 	
ht.insert("S1", 88);//Updating S1 grade
System.out.println("\nHash table after update:");
ht.display();
``
	``
	Hash table after insertions:
	Index 0: A (90) -> S2 (78) -> null
	Index 1: null
	Index 2: null
	Index 3: null
	Index 4: null
	Index 5: null
	Index 6: S1 (85) -> null
	
	Hash table after update:
	Index 0: A (90) -> S2 (78) -> null
	Index 1: null
	Index 2: null
	Index 3: null
	Index 4: null
	Index 5: null
	Index 6: S1 (88) -> None
	``
		For S1:
		'S' = 83, '1' = 49.
		Total = 83 + 49 = 132.
		Index = 132 % 7 = 6.
		For A:
		'A' = 65, 'Y' = 89.
		Total = 65 + 89 = 154.
		Index = 154 % 7 = 0.
		For S2:
		'S' = 83, '2' = 50.
		Total = 83 + 50 = 133.
		Index = 133 % 7 = 0.

32	48 0	64 @	80 P	96 `	112 p
33 !	49 1	65 A	81 Q	97 a	113 q
34 "	50 2	66 B	82 R	98 b	114 r
35 #	51 3	67 C	83 S	99 c	115 s
36 $	52 4	68 D	84 T	100 d	116 t
37 %	53 5	69 E	85 U	101 e	117 u
38 &	54 6	70 F	86 V	102 f	118 v
39 '	55 7	71 G	87 W	103 g	119 w
40 (	56 8	72 H	88 X	104 h	120 x
41)	57 9	73 I	89 Y	105 i	121 y
42 *	58 :	74 J	90 Z	106 j	122 z
43 +	59 ;	75 K	91 [	107 k	123 {
44 ,	60 <	76 L	92 \	108 l	124	
45 -	61 =	77 M	93]	109 m	125 }
46 .	62 >	78 N	94 ^	110 n	126 ~
47 /	63 ?	79 O	95 _	111 o	127 ¨