Exercise:14

class HashTable:

def \_\_init\_\_(self, size):

self.size = size

self.table = [None] \* size

def hash\_function(self, key):

return key % self.size

def insert(self, key):

index = self.hash\_function(key)

start\_index = index

while self.table[index] is not None:

index = (index + 1) % self.size

if index == start\_index:

print("Hash Table is full! Cannot insert key:", key)

return

self.table[index] = key

print(f"Key {key} inserted at index {index}.")

def display(self):

print("Hash Table:")

for i, key in enumerate(self.table):

print(f"Index {i}: {key}")

# Example usage:

if \_\_name\_\_ == "\_\_main\_\_":

ht = HashTable(7)

ht.insert(10)

ht.insert(20)

ht.insert(5)

ht.insert(15)

ht.insert(28)

ht.insert(89)

ht.insert(72)

ht.insert(10)

ht.display()

output:

==

Key 10 inserted at index 3.

Key 20 inserted at index 6.

Key 5 inserted at index 5.

Key 15 inserted at index 1.

Key 28 inserted at index 0.

Key 89 inserted at index 2.

Key 72 inserted at index 4.

Hash Table is full! Cannot insert key: 10

Hash Table:

Index 0: 28

Index 1: 15

Index 2: 89

Index 3: 10

Index 4: 72

Index 5: 5

Index 6: 20