class HashTableChaining:

def \_\_init\_\_(self, size, with\_replacement=False):

self.size = size

self.with\_replacement = with\_replacement

self.table = [[] for \_ in range(size)]

def hash\_function(self, key):

return hash(key) % self.size

def insert(self, key, value):

index = self.hash\_function(key)

if not self.with\_replacement:

for pair in self.table[index]:

if pair[0] == key:

return False # Key already exists, insertion failed

self.table[index].append((key, value))

return True # Insertion successful

def find(self, key):

index = self.hash\_function(key)

for pair in self.table[index]:

if pair[0] == key:

return pair[1] # Return value associated with the key

return None # Key not found

def delete(self, key):

index = self.hash\_function(key)

for i, pair in enumerate(self.table[index]):

if pair[0] == key:

del self.table[index][i]

return True # Deletion successful

return False # Key not found, deletion failed

def input\_record():

key = input("Enter Key:")

value = input("Enter Value:")

return key, value

def main():

choice1 = 0

size = int(input("Enter the size of the hash table: "))

choice2 = int(input("Choose collision handling method:\n1. Without Replacement\n2. With Replacement\nEnter choice: "))

if choice2 == 1:

dictionary = HashTableChaining(size, with\_replacement=False)

elif choice2 == 2:

dictionary = HashTableChaining(size, with\_replacement=True)

else:

print("Invalid choice. Exiting...")

return

while choice1 != 4:

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("1. Insert")

print("2. Find")

print("3. Delete")

print("4. Exit")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

choice1 = int(input("Enter choice: "))

if choice1 == 1:

key, value = input\_record()

if dictionary.insert(key, value):

print("Insertion successful")

else:

print("Insertion failed. Key already exists.")

elif choice1 == 2:

key = input("Enter Key to find:")

result = dictionary.find(key)

if result is not None:

print("Value found:", result)

else:

print("Key not found")

elif choice1 == 3:

key = input("Enter Key to delete:")

if dictionary.delete(key):

print("Deletion successful")

else:

print("Key not found. Deletion failed.")

elif choice1 == 4:

print("Exiting...")

break

else:

print("Invalid choice. Please try again.")

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

main()