**Code**

class Node:

def \_\_init\_\_(self, key, value):

self.key = key

self.value = value

self.next = None

def \_\_str\_\_(self):

return f"({self.key}, {self.value})"

class HashTable:

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

self.size = size

self.table = [None] \* size

self.replacement = replacement

def hash(self, key):

return hash(key) % self.size

def insert(self, key, value):

index = self.hash(key)

if self.table[index] is None:

self.table[index] = Node(key, value)

else:

if self.replacement:

node = self.table[index]

while node.next is not None and node.key != key:

node = node.next

if node.key == key:

node.value = value

else:

node.next = Node(key, value)

else:

node = self.table[index]

while node.next is not None:

node = node.next

node.next = Node(key, value)

def find(self, key):

index = self.hash(key)

node = self.table[index]

while node is not None:

if node.key == key:

return node.value

node = node.next

return None

def delete(self, key):

index = self.hash(key)

node = self.table[index]

prev\_node = None

while node is not None:

if node.key == key:

if prev\_node is None:

self.table[index] = node.next

else:

prev\_node.next = node.next

return

prev\_node = node

node = node.next

def display(self):

for i in range(self.size):

print(f"Bucket {i}: ", end="")

node = self.table[i]

while node is not None:

print(node, end=" -> ")

node = node.next

print("None")

# Example usage with user input

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

replacement = input("Do you want to use replacement? (y/n): ").lower() == 'y'

ht = HashTable(size, replacement)

while True:

print("1. Insert")

print("2. Find")

print("3. Delete")

print("4. Display")

print("5. Exit")

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

if choice == 1:

key = input("Enter key: ")

value = input("Enter value: ")

ht.insert(key, value)

print("Key-value pair inserted.")

elif choice == 2:

key = input("Enter key: ")

value = ht.find(key)

if value is not None:

print(f"Value for key {key}: {value}")

else:

print(f"Key {key} not found.")

elif choice == 3:

key = input("Enter key: ")

ht.delete(key)

print("Key-value pair deleted.")

elif choice == 4:

ht.display()

elif choice == 5:

break

else:

print("Invalid choice.")