```
class Node:
    def __init__(self, key, value):
        self.key = key
        self.value = value
        self.next = None


class HashTable:
    def __init__(self, size):
        self.table = [None] * size

    def hash(self, key):
        return key % len(self.table)

    def insert(self, key, value):
        index = self.hash(key)
        node = Node(key, value)
        if self.table[index] is None:
            self.table[index] = node
        else:
            current = self.table[index]
            while current.next is not None:
                current = current.next
            current.next = node

    def search(self, key):
        index = self.hash(key)
        current = self.table[index]
        while current is not None:
            if current.key == key:
                return current.value
            current = current.next
        return None

    def delete(self, key):
        index = self.hash(key)
        current = self.table[index]
        previous = None
        while current is not None:
            if current.key == key:
                if previous is None:
                    self.table[index] = current.next
                else:
                    previous.next = current.next
                return
            previous = current
            current = current.next


# Example usage:

hash_table = HashTable(10)
hash_table.insert(5, 6)
hash_table.insert(5, 8)
print(hash_table.search(5))  # 6
hash_table.delete(5)
print(hash_table.search(5))  # None
```