```
class HashTable:
    def __init__(self, size):
        self.size = size
        self.table = [[] for _ in range(self.size)]

    def hash_function(self, key):
        return key % self.size

    def insert(self, key, value):
        hash_index = self.hash_function(key)

        # Check if key already exists; update if found
        for i, (existing_key, _) in enumerate(self.table[hash_index]):
            if existing_key == key:
                self.table[hash_index][i] = (key, value)
                print(f"Key {key} updated with new value {value}")
                return

        # If key not found, append new key-value pair
        self.table[hash_index].append((key, value))
        print(f"Key {key} with value {value} inserted")

    def search(self, key):
        hash_index = self.hash_function(key)

        for existing_key, value in self.table[hash_index]:
            if existing_key == key:
                print(f"Value for key {key}: {value}")
                return value
        print(f"Key {key} not found")
        return None

    def delete(self, key):
        hash_index = self.hash_function(key)

        for i, (existing_key, _) in enumerate(self.table[hash_index]):
            if existing_key == key:
                del self.table[hash_index][i]
                print(f"Key {key} deleted")
                return True

        print(f"Key {key} not found for deletion")
        return False

    def display(self):
        print("\nHash Table Contents:")
        for i, chain in enumerate(self.table):
            print(f"Table {i+1}: {chain}")


# Example usage:
if __name__ == "__main__":
    hash_table = HashTable(7)

    while True:
        print("\nHash Table Operations:")
        print("1. Insert")
        print("2. Search")
        print("3. Delete")
        print("4. Display")
        print("5. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            key = int(input("Enter key (integer): "))
            value = input("Enter value: ")
            hash_table.insert(key, value)
        elif choice == '2':
            key = int(input("Enter key to search: "))
            hash_table.search(key)
        elif choice == '3':
            key = int(input("Enter key to delete: "))
            hash_table.delete(key)
        elif choice == '4':
            hash_table.display()
        elif choice == '5':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Please try again.")
```