PIMPRI CHINCHWAD EDUCATION TRUST's.
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
(An Autonomous Institute)

Class : SY BTech **Acad. Yr. 2025-26** **Semester : I**

Name of the student: Hariom Shrikrishna Gundale **PRN : 124B1B036**

Department: Computer Engineering **Division : A**

Course Name : Data Structures Laboratory **Course Code:BCE23PC02**

Completion Date : 15/10/2025

Assignment No. 10

Problem Statement: Write a C++ Program to insert elements in Hash Table using Separate Chaining.

Source Code :

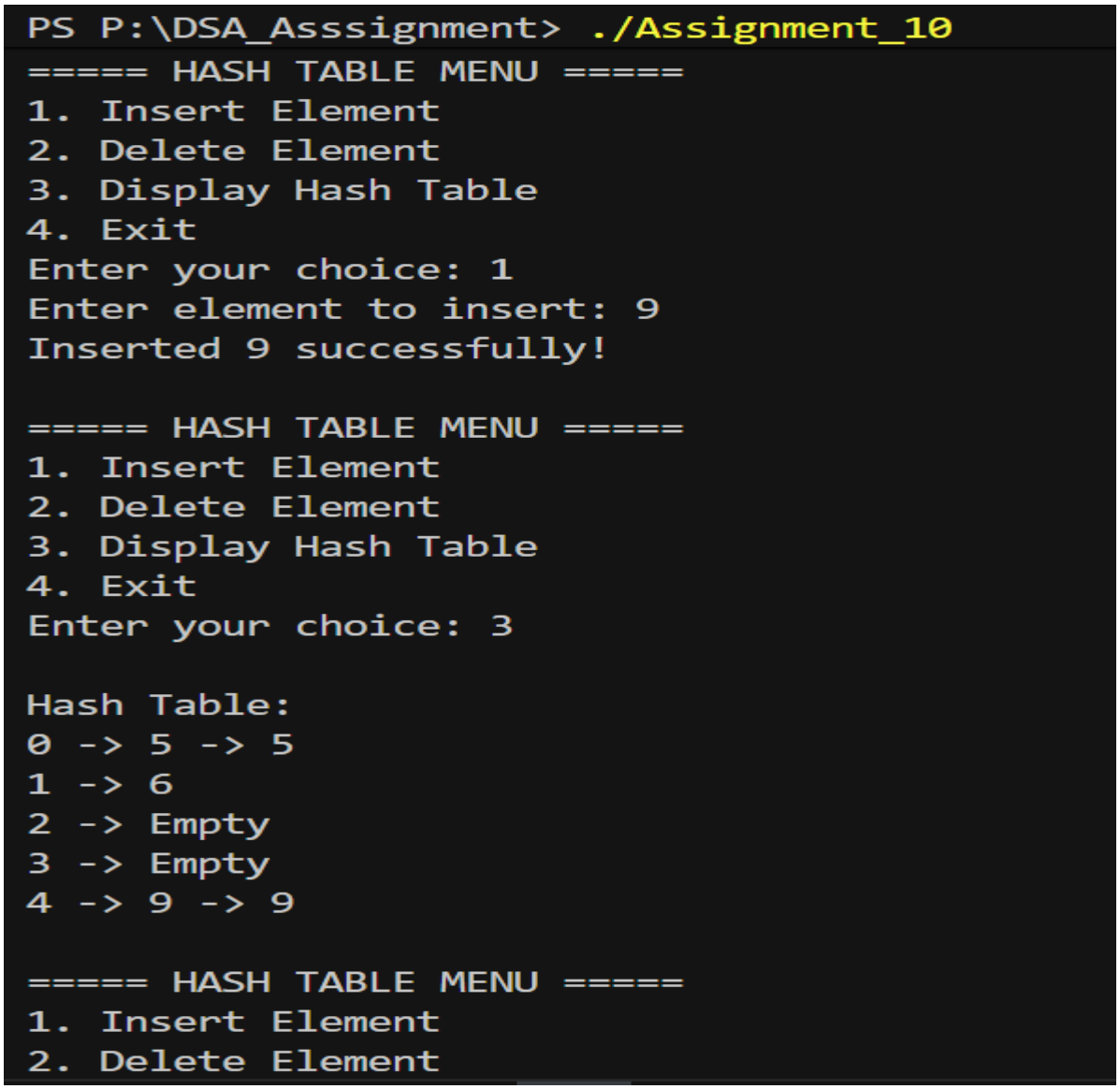
```
#include <iostream>
using namespace std;

class Node {
public:
    int data;
    Node* next;

    Node(int val) {
        data = val;
        next = nullptr;
    }
};

class HashTable {
private:
    int size;
    Node** table;

    int hashFunction(int key) {
        return key % size;
    }
```

```
public:

    HashTable(int s) {
        size = s;
        table = new Node*[size];
        for (int i = 0; i < size; i++)
            table[i] = nullptr;
    }

    void insert(int key) {
        int index = hashFunction(key);
        Node* newNode = new Node(key);

        if (table[index] == nullptr) {
            table[index] = newNode;
        } else {
            Node* temp = table[index];
            while (temp->next != nullptr)
                temp = temp->next;
            temp->next = newNode;
        }
        cout << "Inserted " << key << " successfully!\n";
    }

    void remove(int key) {
        int index = hashFunction(key);
        Node* temp = table[index];
        Node* prev = nullptr;

        while (temp != nullptr && temp->data != key) {
            prev = temp;
            temp = temp->next;
        }

        if (temp == nullptr) {
            cout << "Key " << key << " not found!\n";
            return;
        }

        if (prev == nullptr)
            table[index] = temp->next;
        else
            prev->next = temp->next;

        delete temp;
        cout << "Deleted " << key << " successfully!\n";
    }

    void display() {
        cout << "\nHash Table:\n";
```

```
        for (int i = 0; i < size; i++) {
            cout << i << " -> ";
            Node* temp = table[i];
            if (temp == nullptr)
                cout << "Empty";
            while (temp != nullptr) {
                cout << temp->data;
                if (temp->next != nullptr)
                    cout << " -> ";
                temp = temp->next;
            }
            cout << endl;
        }
    }

    ~HashTable() {
        for (int i = 0; i < size; i++) {
            Node* temp = table[i];
            while (temp != nullptr) {
                Node* del = temp;
                temp = temp->next;
                delete del;
            }
        }
        delete[] table;
    }
};

int main() {
    int size, choice, key;
    cout << "Enter number of buckets in hash table: ";
    cin >> size;

    HashTable h(size);

    while (true) {
        cout << "\n===== HASH TABLE MENU =====\n";
        cout << "1. Insert Element\n";
        cout << "2. Delete Element\n";
        cout << "3. Display Hash Table\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            cout << "Enter element to insert: ";
            cin >> key;
            h.insert(key);
            break;
```

```
        case 2:
            cout << "Enter element to delete: ";
            cin >> key;
            h.remove(key);
            break;

        case 3:
            h.display();
            break;

        case 4:
            cout << "Exiting program...\n";
            return 0;

        default:
            cout << "Invalid choice! Please try again.\n";
        }
    }

    return 0;
}
```

Screen Shot of Output :



Conclusion:

Thus, we have successfully implemented the C++ Program to insert elements in Hash Table using Separate Chaining.