ASSIGNMENT NO.11

NAME : PAYAL BHAGVAN CHARVANDE

PRN NO.: 125B2F002

CLASS: SY DIV: IT A

CODE:

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

// Structure for Student
struct Student {
   int id;
   string name;
   int age;
   float marks;
   Student* next; // for handling collisions (chaining)
};
// Define hash table size
const int SIZE = 10;

// Hash Table (Array of pointers)
Student* hashTable[SIZE];

// Hash function (simple modulo)
int hashFunction(int id) {
```

```
    return id % SIZE;
}
// Function to add a new student
void addStudent(int id, string name, int age, float marks) {
    int index = hashFunction(id);
    Student* newStudent = new Student;
    newStudent->id = id;
    newStudent->name = name;
    newStudent->age = age;
    newStudent->marks = marks;
    newStudent->next = NULL;

    // If slot is empty
    if (hashTable[index] == NULL) {
        hashTable[index] = newStudent;
    } else {
        // Collision → insert at end of linked list
        Student* temp = hashTable[index];
        while (temp->next != NULL) {
            if (temp->id == id) {
                cout << "Student ID already exists!\n";
                delete newStudent;
                return;
            }
            temp = temp->next;
        }
        temp->next = newStudent;
    }
```

```
    cout << "Student record added successfully!\n";
}
// Function to search a student by ID
void searchStudent(int id) {
    int index = hashFunction(id);
    Student* temp = hashTable[index];

    while (temp != NULL) {
        if (temp->id == id) {
            cout << "\nStudent Found!\n";
            cout << "ID: " << temp->id << endl;
            cout << "Name: " << temp->name << endl;
            cout << "Age: " << temp->age << endl;
            cout << "Marks: " << temp->marks << endl;
            return;
        }
        temp = temp->next;
    }
    cout << "Student with ID " << id << " not found!\n";
}
// Function to delete a student by ID
void deleteStudent(int id) {
    int index = hashFunction(id);
    Student* temp = hashTable[index];
    Student* prev = NULL;

    while (temp != NULL) {
        if (temp->id == id) {
```

```
            if (prev == NULL)
                hashTable[index] = temp->next;
            else
                prev->next = temp->next;
            delete temp;
            cout << "Student deleted successfully!\n";
            return;
        }
        prev = temp;
        temp = temp->next;
    }
    cout << "Student not found!\n";
}
// Function to display all students
void displayAll() {
    cout << "\n--- Student Records ---\n";
    for (int i = 0; i < SIZE; i++) {
        Student* temp = hashTable[i];
        if (temp != NULL) {
            cout << "Bucket " << i << ": ";
            while (temp != NULL) {
                cout << "[ID: " << temp->id << ", Name: " << temp->name
                    << ", Age: " << temp->age << ", Marks: " << temp->marks << "] -> ";
                temp = temp->next;
            }
            cout << "NULL\n";
        }
    }
```

```
}
int main() {
    // Initialize hash table
    for (int i = 0; i < SIZE; i++)
        hashTable[i] = NULL;

    int choice, id, age;
    float marks;
    string name;

    do {
        cout << "\n=== Student Information System (Hashing) ===\n";
        cout << "1. Add Student Record\n";
        cout << "2. Search Student by ID\n";
        cout << "3. Delete Student Record\n";
        cout << "4. Display All Records\n";
        cout << "0. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1:
            cout << "Enter Student ID: ";
            cin >> id;
            cout << "Enter Name: ";
            cin.ignore();
            getline(cin, name);
            cout << "Enter Age: ";
```

```
            cin >> age;
            cout << "Enter Marks: ";
            cin >> marks;
            addStudent(id, name, age, marks);
            break;
        case 2:
            cout << "Enter Student ID to search: ";
            cin >> id;
            searchStudent(id);
            break;
        case 3:
            cout << "Enter Student ID to delete: ";
            cin >> id;
            deleteStudent(id);
            break;
        case 4:
            displayAll();
            break;
        case 0:
            cout << "Exiting program...\n";
            break;
        default:
            cout << "Invalid choice! Try again.\n";
        }
    } while (choice != 0);
    return 0;
}
```

OUTPUT:

```
=== Student Information System (Hashing) ===
1. Add Student Record
2. Search Student by ID
3. Delete Student Record
4. Display All Records
0. Exit
Enter your choice: 1
Enter Student ID: 101
Enter Name: PAYAL
Enter Age: 18
Enter Marks: 88
Student record added successfully!

=== Student Information System (Hashing) ===
1. Add Student Record
2. Search Student by ID
3. Delete Student Record
4. Display All Records
0. Exit
Enter your choice: 2
Enter Student ID to search: 101

Student Found!
ID: 101
Name: PAYAL
Age: 18
Marks: 88

=== Student Information System (Hashing) ===
1. Add Student Record
2. Search Student by ID
3. Delete Student Record
4. Display All Records
0. Exit
Enter your choice: 3
Enter Student ID to delete: 101
Student deleted successfully!
```