```cpp
//HASH TABLE

#include <iostream>

using namespace std;

struct Node {
    int key;
    Node* next;
};

struct HashTable {
    Node** head;
    int size;
};

HashTable createHashTable(int size = 0) {
    HashTable newTable;
    newTable.size = size;
    newTable.head = new Node * [size];

    for (int i = 0; i < size; i++) newTable.head[i] = nullptr;

    return newTable;
}

int calculateHash(int id) {
    return id % 10;
}

void insertTableNode(Node*& head, int key) {
    head = new Node{ key, head };
}

void insertItem(HashTable& table, int key) {
    int index = calculateHash(key);
    insertTableNode(table.head[index], key);
}

void deleteItem(HashTable& table, int key) {
    int index = calculateHash(key);
    Node* prev = nullptr, * cur = table.head[index];
    while (cur && cur->key != key) {
        prev = cur, cur = cur->next;
    }

    if (!cur) return;

    if (!prev) {
        table.head[index] = table.head[index]->next;
    }
    else {
        prev->next = cur->next;
    }
    delete cur;
}

void displayHashTable(HashTable table) {
    for (int i = 0; i < table.size; i++) {
        cout << i << " -> ";
        for (Node* p = table.head[i]; p; p = p->next) {
            cout << p->key << " ";
        }
        cout << endl;
    }
}
```