Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: DSC (01CT0308)	**Aim:** Implementation of hashing functions with different collision resolution techniques.	
Experiment No: 10	**Date:** **26- 10 - 2023**	**Enrolment No:-** 92200133030

Experiment – 10

Objective: Implementation of hashing functions with different collision resolution techniques.

Code :-

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

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

class HashTable {
private:
    vector<Node*> table;
    int tableSize;

    int hash(int key) {
        return key % tableSize;
    }

public:
    HashTable(int size) {
        tableSize = size;
        table.resize(size, NULL);
    }

    void insert(int key) {
        int index = hash(key);
        Node* newNode = new Node;
        newNode->data = key;
```

```
    newNode->next = NULL;

    if (table[index] == NULL) {
        table[index] = newNode;
    } else {
        Node* current = table[index];
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}

bool search(int key) {
    int index = hash(key);
    Node* current = table[index];
    while (current != NULL) {
        if (current->data == key) {
            return true;
        }
        current = current->next;
    }
    return false;
}

void remove(int key) {
    int index = hash(key);
    Node* current = table[index];
    Node* prev = NULL;

    while (current != NULL && current->data != key) {
        prev = current;
        current = current->next;
    }

    if (current == NULL) {
        return;
    }

    if (prev == NULL) {
        table[index] = current->next;
    } else {
        prev->next = current->next;
    }

    delete current;
}
```

```
};

int main() {
    HashTable hashTable(10);

    hashTable.insert(10);
    hashTable.insert(20);
    hashTable.insert(30);

    cout << "Search 20: " << (hashTable.search(20) ? "Found" : "Not Found") << endl;

    hashTable.remove(20);
    cout << "Search 20: " << (hashTable.search(20) ? "Found" : "Not Found") << endl;

    return 0;
}
```

Output:

```
Search 20: Found
Search 20: Not Found

--------------------------------
Process exited after 0.02644 seconds with return value 0
Press any key to continue . . .
```