

Name of Student :- Pavan Ratan Gole

UID:- Na

Batch:- C4

Exp. No. 9

Aim:- Implement the hash concept by hashing the given 'n' keys using the modulo division method and solve the collision using chaining.

Program:-

```
package Java;

/**
 * Chain
 */

class Node {
    private Node head = null;
    int data;
    Node next = null;

    Node(int data) {
        this.data = data;
        this.head = this;
    }

    Node createNode(int data) {
        head = this;
        Node node = new Node(data);
        if(head == null) {
            head = node;
        }
        Node temp = head;
        while(temp.next != null) {
            temp = temp.next;
        }
    }
}
```

```

    }

    temp.next = node;
    return head;
}

Node deleteNode(int data) {
    head = this;
    Node temp = searchNode(data);
    if(temp != null) {
        if(temp == head) {
            head = head.next;
            return head;
        }
        Node cur = head;
        while(cur != null) {
            if(cur.next == temp) {
                break;
            }
            cur = cur.next;
        }
        if(temp.next == null) {
            cur.next = null;
            return head;
        }
        else {
            cur.next = temp.next;
            return head;
        }
    }
    return head;
}

Node searchNode(int data) {
    head = this;
    Node temp = head;
    while(temp != null) {
        if(temp.data == data) {
            return temp;
        }
    }
}

```

```

        temp = temp.next;
    }
    return null;
}

static void display(Node node) {
    while(node != null) {
        System.out.print( " ----> " + node.data);
        node = node.next;
    }
}
}

public class Chain {

    static Node[] arr = new Node[10];
    public static void main(String[] args) {
        for (int i = 0; i < arr.length; i++) {
            insert(i);
            insert(10 + i);
        }
        for (int i = 0; i < arr.length; i++) {
            display(i);
        }
        System.out.println("=====");
        System.out.println("Inserting 21");
        System.out.println("=====");
        insert(21);
        display(1);
        System.out.println("=====");
        System.out.println("Deleting 21");
        System.out.println("=====");
        delete(21);
        display(1);
        System.out.println("=====");
        System.out.println("Searching 19");
        System.out.println("=====");
        search(19);
    }
}

```

```

}

static boolean checkCollision(int index) {
    if(arr[index] != null) {
        return true;
    }
    return false;
}

static void insert(int data) {
    int index = data % 10;
    if(checkCollision(index)) {
        arr[index].createNode(data);
    }
    else {
        arr[index] = new Node(data);
    }
}

static void display(int index) {
    System.out.print(index + " ");
    Node.display(arr[index]);
    System.out.println();
}

static void delete(int data) {
    int index = data % 10;
    Node temp = arr[index].deleteNode(data);
    arr[index] = temp;
}

static void search(int data) {
    int index = data % 10;
    if(arr[index].searchNode(data) != null) {
        System.out.println("Data Found");
    }
    else {
        System.out.println("Data Not Found");
    }
}
}

```

```
}
```

Output:-

```
0  -> 0  -> 10
1  -> 1  -> 11
2  -> 2  -> 12
3  -> 3  -> 13
4  -> 4  -> 14
5  -> 5  -> 15
6  -> 6  -> 16
7  -> 7  -> 17
8  -> 8  -> 18
9  -> 9  -> 19

=====
Inserting 21
=====
1  -> 1  -> 11 -> 21
=====
```

I was inserting 21 in the Array. As there is a collision happening I used the linked List to resolve the collision. I used for loop to insert the elements in the array of size 10. Used hash function $\text{index} = \text{data} \% 10$;

```
Deleting 21
=====
1  ----> 1  ----> 11
=====
```

I deleted 21 from the array index 0.

```
Searching 19
=====
Data Found
```

Search for 19 in the array and found it.