

Lab#11 Hashing

1. Data Structure

```
class Node { private: int data; Node *link; friend class Htable; };
```

2. ADT 함수 : 1) findkey, 2) insertKey 3) deleteKey 4) printable

3. 메뉴로 구성할 것 (1. Find, 2. Insert 3. Delete 4. Quit)

4. Hashing Function: division method ex) Print Table: table size 7

5. Algorithm : Chaining Method

● Main Program:

```
do { Enter command (i,f,d,q): read(ch);
    if (ch != 'q') {
        print("Enter key "); read(inkey);
        switch (ch){
            case 'i': check=insertkey(inkey);
                if (check==false) print("Cannot insert key"); break;
            case 'f': check=findkey(inkey, index);
                if (check==false) print("key not found"); break;
            case 'd': check=deletekey(inkey);
                if (check==false) print("key not found "); break;
            default : print("Bad Command");
        }
        printtable(); // insert, delete, find 후에 매번 테이블의 내용출력.
    }while (ch == 'q');
```

● FindKey

- Get index value for Key
- Get head node from HashTable
- If (headnode != NULL) {
- Search the table for the Key // q= head; q= q->link
- If (q = null) return false;
- Else return true }
- Else return false;

- Insert Key

- Get Index for the Key
- Check = findkey(key);
- If (check == true) return false;
- Get head node for the Key
- If (head node = NULL) insert new node
- Else find place to insert the key // singly linked list 와 유사함

- Delete Key

- Get index for the key
- Get head node for the Key
- If (head = key) delete head node & move head= head->next; (update head node)
- Else {
- Find node and delete the node for the key // same as singly linked list

- Print Table

```
For(i= 0; i < maxsize; i++) {  
    Print "Htable [i]"  
    Get head node for "i"  
    For (head; head!= NULL; head= head->link)  
        Print "node"
```

- Testing 예):

Insert 7→insert 36 → insert 29→insert 22 →find 25 → delete 22 →insert 8 → insert 17
Insert ->10→delete 17→insert 3

Htable[0]: 7

Htable[1]: 36 →29→ 8

Htable[2]:

Htable[3]: 10→3

Htable[4]:

Htable[5]:

Htable[6]: