

# 프로젝트 4: 체이닝 해시 테이블

## ■ 체이닝 해시 테이블 구현하기

- 해시 테이블 크기  $m = 8$ , 키 개수  $n = 9$
- 해시 함수  $h(x) = x \% m$
- 출력
  - ✓ 검색한 데이터 출력
  - ✓ 아래와 같이, 해시 값 별로 체이닝 되어 있는 데이터 출력
- LMS “Project4\_Chaining\_Hash\_Table”
  - ✓ 제출물: 소스 코드 및 실행 화면 캡처파일 업로드
  - ✓ 제출기한: ~5/1(수) 23:59

Microsoft Visual Studio 디버그 콘솔

```
ID: 20171513    NAME: YoojinChoi
Hash Table
ID(0): (20171496, EunsoIAn)
ID(1): (20171513, YoojinChoi)
ID(2): (20171514, EuiyeonCho)->(20171506, YoojinKim)->(20171498, JiwooChoi)
ID(3): (20171515, SeungwooBaek)->(20171507, ChanggiLee)
ID(4):
ID(5):
ID(6): (20171518, IsuJung)
ID(7):
```

# 프로젝트 4: 체이닝 해시 테이블

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define MAX_NAME_LEN 20
#define HASH_TABLE_SIZE 8
#define MAX_STUDENT_SIZE 9

typedef struct {
    int id;
    char name[MAX_NAME_LEN];
} Student;

typedef struct node {
    Student person;
    struct node *next;
} Node;

Node *HashTable[HASH_TABLE_SIZE];

void insert(int key, Student A) {
    // key 값에 따라 A 데이터를 해시 테이블에 삽입
}

void delete(int key) {
    // key 값을 갖는 데이터를 해시 테이블에서 삭제
}

Student *search(int key) {
    // key 값을 갖는 데이터를 찾아서 해당 데이터의 포인터를 반환
    // key 값을 갖는 데이터가 없을 경우, NULL을 반환
}
```

# 프로젝트 4: 체이닝 해시 테이블

```
void printHashTable() {  
    // 출력 결과 화면처럼 출력하는 함수 작성  
}  
  
int main() {  
    Student *ptr;  
    Student data[MAX_STUDENT_SIZE] = {  
        {20171496, "EunsolAn"},  
        {20171498, "JiwooChoi"},  
        {20171506, "YoojinKim"},  
        {20171507, "ChanggiLee"},  
        {20171513, "YoojinChoi"},  
        {20171514, "EuiyeonCho"},  
        {20171515, "SeungwooBaek"},  
        {20171518, "IsuJung"},  
        {20171540, "SanghoonSong"} };  
  
    for (int i = 0; i < MAX_STUDENT_SIZE; i++)  
        insert(data[i].id, data[i]);  
  
    ptr = search(20180345);  
    if (ptr != NULL)  
        printf("ID: %d\tNAME: %s\n", ptr->id, ptr->name);  
    ptr = search(20171513);  
    if (ptr != NULL)  
        printf("ID: %d\tNAME: %s\n", ptr->id, ptr->name);  
  
    delete(20171540);  
    printHashTable();  
}
```