**개요**

array와 LinkedList를 이용하여, 이름을 입력하면 구조체에 저장되어 있는 문자열인 학번이 출력되고, 학번을 입력하면, 구조체에 저장되어 있는 문자열인 이름이 출력이 되는 프로그램이다.

**프로그램 구조 설명**

**함수에 대한 설명**

**Array**

void Init(Student\* student)는 이름, 학번을 입력 받기 전, 공간에 값을 초기화 해준다.

void Insert(Student\* student, char\* name, char\* number)는 strcpy를 사용하여 문자열을 복사하여 이름 학번에 저장해준다.

void PPAP(Student\* student)는 배열에 저장한 정보를 프린트해준다.

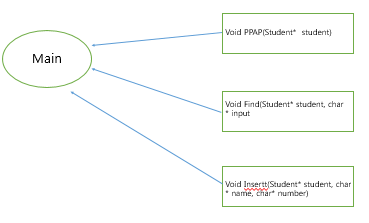
void Find(Student\* student, char\* input)는 마지막으로 학번과 이름을 배열에 저장하였으니, 이름을 입력하면 한번이 그래서 strcmp함수를 사용하여 입력한 값과 학생의 이름이 같다면 0이 출력 값으로 나오니 if문을 사용하여 0이 나오면 그 학생의 학번이 프린트 되게 만들었다. 만약 학번과 이름을 쳤는데, 배열에 없는 값이 나오면 에러라는 걸 보여주기 위해 초기에 int flag=0이라는 변수를 만들어 초기화하였다. 학번과 이름이 같은 경우는 flag=1이라 하고, 만약 for문이 다 돌았는데도 flag값이 바뀌지 않는다면, Not found를 출력하게 하였다.

**LinkedList**

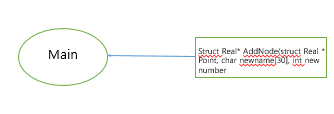
struct Real\* Addnote(struct Real\* Point, char newname[30], int newnumber)는 맨 앞에 void가 아닌 struct Real\*를 쓴 이유는 함수 마지막을 보면 return newNode라고 되어있는데, newNode라는 값을 반환하기 때문이다. 처음 구조체 방을 만들어 주고, 입력 받은 구조체 방의 next값을 처음 구조체 방에 넣어준다. 그리고 입력 받은 이름과 숫자를 처음 구조체 방에 넣어주고, 입력 받은 구조체 방의 next는 처음 구조체 방을 가리키게 한다.

**다이어그램**

Array

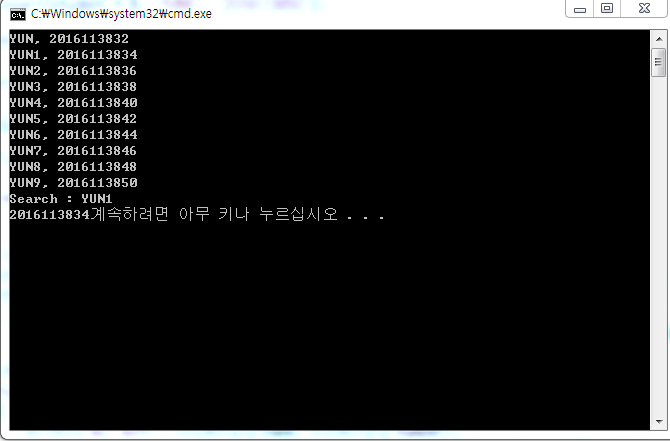


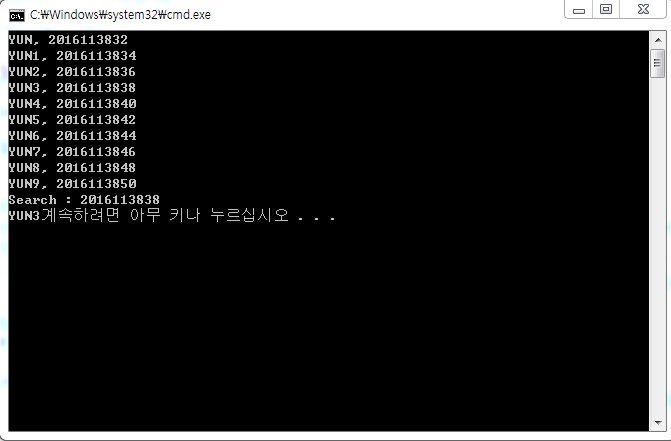
LinkedList



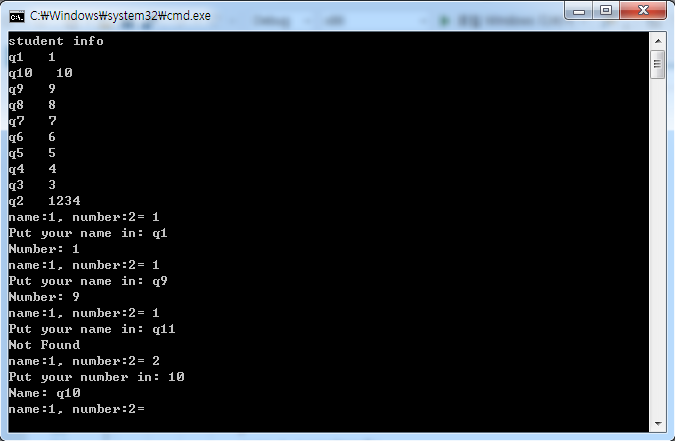
**실행결과**

**array를 사용했는 경우,**



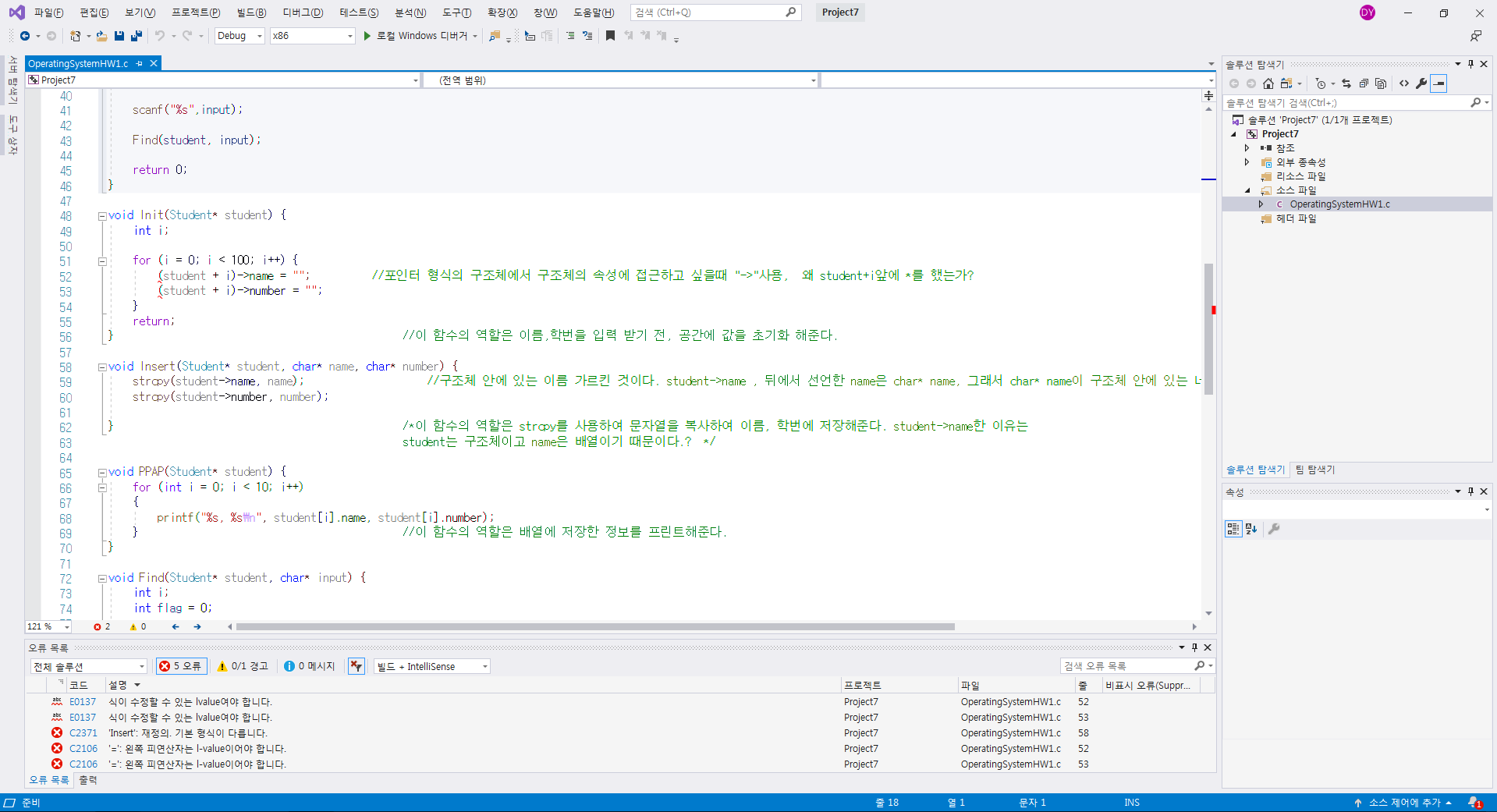


**LinkedList를 사용한 경우,**



**고찰**

배열을 사용하여 코드를 작성할 때, 포인터, 배열, 구조체에 대해 아에 모르는 상태였어서, 친구(권현수)에게 포인터, 배열, 구조체에 대해 배웠습니다. 배우고 난 후에도 그 코드를 어떤 형식으로 짜야 할 지 감이 안와서 친구에게 어떤 흐름으로 코드를 작성해야 할지 들은 후, 직접 짜보았습니다. 흐름을 알게 된 후 코드를 짜니 훨씬 수월했습니다. 처음에 함수를 사용하지 않고, 메인에다가 다 코드를 작성하였는데, 너무 가독성이 떨어져서, 함수를 사용하여 코드를 작성하였습니다. 제가 이 코드를 작성하면서 안 어려웠던 부분은 없었지만, 제일 어려웠던 부분은 포인터입니다. 구조체를 정의하고 값을 입력 받으려고 했지만 컴파일 에러가 계속 떠서, swap이라는 함수를 공부하면서 왜 에러가 뜨는지 알게 되었습니다. 그리고 구조체를 정의하고 예를 들어 메인함수에서 구조체 student라고 하면 구조체가 자료형이 되고 student가 그 자료형의 이름이 된다는 것을 알고 난 후부터 조금 수월하게 코드 작성을 하였습니다.



(student + i) ->name= “”; 초기화를 하는 과정에서 화살표 연산자는 포인터 형식의 구조체에서 구조체의 속성에 접근하고 싶을 때 사용하는 것인데, 맨 앞에 포인터를 쓰지 않아서 에러가 나타났다.

LinkedList를 사용하여 코드를 작성할 땐, 박재성 학생에게 배웠고, 제일 어려웠던 부분이 LinkedList의 정의를 이해하는 것이 너무 힘들어서 코드를 어떻게 작성해야 할지 감이 너무 안 왔었습니다. LinkedList를 할 때, 비교하는 함수로 strcmp를 사용하였는데, 제가 이미 학번은 int로 선언했는데 strcmp으로 비교하려고 해서 에러가 떴습니다. 그래서 ==를 사용하여 이 문제를 해결하였습니다.

**고찰**

저는 이 과제에 시간을 매우 많이 투자하였습니다. 저는 코딩을 1학년이후로 한번도 하지 않아서 scanf하는 것 조차 제대로 할 줄 몰랐습니다. 그래서 매일 권현수 학생과 저녁9시에 만나서 새벽3~4시까지 이 과제를 하는데 필요한 c언어 지식을 쌓아갔고, 그 결과 코딩을 다 만들었습니다. 하지만 아직 어떤 task가 주어지면, 어떻게 코딩을 할 지 감이 오질 않습니다. 그리고 pointer와 linkedlist에 대해 완벽하게 이해를 하진 못하였습니다. 이 부분에 대해서는 제가 공부를 더하고 교수님께 배워야 할 것 같습니다. 이번 학기 정말 열심히 코딩 공부해 볼 생각입니다!

**프로그램 소스파일**

**Array**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h> //동적 메모리 할당 malloc함수를 사용하기 위해서 헤더파일을 추가하였다.

#include <string.h> //strcpy, strcmp함수를 사용하기 위해서 헤더파일을 추가하였다.

typedef struct Student {

char name[30];

char number[11];

}Student; //처음 구조체를 정의하여 이름 학번을 입력받을 수 있는 공간을 만들어 주었다. 배열=포인터, 다른 함수 파라미터로 받아올려면 포인터 형식으로. (like swap 함수)

void PPAP(Student\* student);

void Find(Student\* student, char\* input);

void Insert(Student\* student, char\* name, char\* number);

int main() {

Student student [100]; //구조체 100개를 만들어 주었다.

char\* input;

input = (char\*)malloc(sizeof(char) \* 1000);

Insert(student, "YUN", "2016113832"); //58번 줄

Insert(student+1, "YUN1", "2016113834");

Insert(student + 2, "YUN2", "2016113836");

Insert(student + 3, "YUN3", "2016113838");

Insert(student + 4, "YUN4", "2016113840");

Insert(student + 5, "YUN5", "2016113842");

Insert(student + 6, "YUN6", "2016113844");

Insert(student + 7, "YUN7", "2016113846");

Insert(student + 8, "YUN8", "2016113848");

Insert(student + 9, "YUN9", "2016113850");

PPAP(student); //65번 줄

printf("Search : ");

scanf("%s",input);

Find(student, input);

return 0;

}

void Insert(Student\* student, char\* name, char\* number) {

strcpy(student->name, name); //구조체 안에 있는 이름 가르킨 것이다. student->name , 뒤에서 선언한 name은 char\* name, 그래서 char\* name이 구조체 안에 있는 네임에 들어가는 것이다.

strcpy(student->number, number);

} //이 함수의 역할은 strcpy를 사용하여 문자열을 복사하여 이름, 학번에 저장해준다.

void PPAP(Student\* student) {

for (int i = 0; i < 10; i++)

{

printf("%s, %s\n", student[i].name, student[i].number);

} //이 함수의 역할은 배열에 저장한 정보를 프린트해준다.

}

void Find(Student\* student, char\* input) {

int i;

int flag = 0;

for (i = 0; i < 10; i++) {

if (strcmp(student[i].name, input) == 0) {

printf("%s", student[i].number);

flag = 1;

}

else if (strcmp(student[i].number, input) == 0) {

printf("%s", student[i].name);

flag = 1;

}

}

if (flag == 0) {

printf("Not Found!");

}

}

/\*마지막으로 학번과 이름을 배열에 저장하였으니, 이름을 입력하면 학번이 나와야 하고 학번을 입력하면 이름이 나와야 한다. 그래서 strcmp함수를 사용하여 입력한 값과 학생의 이름이 같다면 0이 출력 값으로 나오니 if문을 사용하여 0이 나오면 그 학생의 학번이 프린트 되게 만들었다. 만약 학번과 이름을 쳤는데 배열에 없는 값이 나오면 에러라는 걸 보여주기 위해 초기에 int flag=0라는 변수를 만들어 초기화를 하고, 학번과 이름이 있다면 flag=1을 하여 flag값을 바꿔주고, 만약 for문이 다 돌았는데도 flag값이 바뀌지 않는다면,Not found를 출력하게 하였다.\*/

**LinkedList**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Real {

struct Real\* next; //다음 노드 주소 저장 포인터.

char name[30];

int number;

};

struct Real\* AddNode(struct Real\* Point, char newname[30], int newnumber) //기준점 누구뒤로 붙을건지, 그리고 정보

{

struct Real\* newNode; //new node라는 노드가 한개 더 생기고

newNode = malloc(sizeof(struct Real)); //

newNode->next = Point->next; //정보를 저장할 노드에 새로 올 노드를 저장.

strcpy(newNode->name, newname); //strcpy가 아닌 strcat을 사용하여서 에러가 계속 났었다. 새로운 노드에 정보 입력

newNode->number = newnumber; //새로운 노드에 정보 입력

Point->next = newNode;

return newNode; //return newNode를 하여 tail, 마지막 기준점을 정해줬다.

}

int main() {

struct Real\* head = malloc(sizeof(struct Real));

head->next = NULL; //head의 next에 nill을 넣어준다음.

struct Real\* temp = malloc(sizeof(struct Real));

temp = AddNode(head, "q1", 1); //첫번째 돌려서 생성된 newNode정보를 temp에 저장해놓고

AddNode(temp, "q2", 1234); //두번째 돌렸을 때 마지막정보를 넣어준다.

AddNode(temp, "q3", 3);

AddNode(temp, "q4", 4);

AddNode(temp, "q5", 5);

AddNode(temp, "q6", 6);

AddNode(temp, "q7", 7);

AddNode(temp, "q8", 8);

AddNode(temp, "q9", 9);

AddNode(temp, "q10", 10);

struct Real\* curr = head->next;

printf("student info \n");

while (curr != NULL) //addnode를 사용하여 넣어준 name과 number를 프린트하였다.

{

printf("%s %d\n", curr->name, curr->number);

curr = curr->next;

}

while (1) { //while 1를 사용하여 계속 돌린다.

int input = 0;

int flag = 0;

printf("name:1, number:2= ");

scanf("%d", &input);

if (input == 1) { //input에 1번이 들어가면 name을 입력받고, strcmp를 사용하여 char 비교후 맞으면 그 노드의 number출력.

char StudentName[20];

printf("Put your name in: ");

scanf("%s", &StudentName);

struct Real\* curr = head->next;

while (curr != NULL)

{

if (strcmp(curr->name, StudentName) == 0) {

printf("Number: %d \n", curr->number);

flag = 1;

break;

}

curr = curr->next;

}

if (flag == 0) { //while (curr != NULL)이 끝까지 갔는데도 flag가 0이면 이름은 없다.

printf("Not Found\n");

}

}

if (input == 2) { //input에 2번이 들어가면 number을 입력받고, ==연산자를 이용하여 비교후 맞으면 그 노드의 name출력.

int SearchNumber;

printf("Put your number in: ");

scanf("%d", &SearchNumber);

struct Real\* curr = head->next;

while (curr != NULL)

{

if (curr->number == SearchNumber) {

printf("Name: %s \n", curr->name);

flag = 1;

break;

}

curr = curr->next;

}

if (flag == 0) {

printf("Not Found\n");

}

}

}

return 0;

}

**자료 출처**

<https://blog.naver.com/zlatmgpdjtiq?Redirect=Log&logNo=221661570066>

<https://blog.naver.com/zerogeas05?Redirect=Log&logNo=221617285690>