

## 고급 C프로그래밍 및 실습 프로젝트 #2

### 연락처 프로그램 2단계

수업 시간에 설명된 설계를 바탕으로 연락처 프로그램(2단계)을 작성한다. 기본적으로 2단계는 1단계의 기능과 동일한 기능을 구현하지만, 다른 자료구조 (구조체 포인터 배열)를 사용하는 것에 중점을 두고 있다.

#### ◎ 연락처에 저장할 자료

연락처에 저장할 자료는 1단계와 동일하나 자료를 저장하고 유지하는 방법이 다른 방식을 사용한다. 한 사람의 정보를 구조체로 표현하는 것은 이전과 동일하지만, 여러 사람의 정보를 저장하기 위해서 구조체 배열 대신, 구조체 포인터 배열을 사용하고, 구조체 내부멤버들도 배열을 사용하지 않고 포인터를 사용한다. 아래와 같이 2단계 방법으로 변수를 선언해야 한다.

<이전 방법>

```
struct 구조체이름 {
    char 이름변수[21];
    char 전화번호변수[16];
    char 생일변수[9];
};
struct 구조체이름 구조체변수이름[100];
```

< 2단계 방법>

```
struct 구조체이름 {
    char *이름변수;
    char *전화번호변수;
    char *생일변수;
};
struct 구조체이름 **구조체변수이름;    // (주의) 이중포인터
```

몇 개의 정보를 처리할 것인가를 프로그램 시작 제일 처음 입력을 받아, 구조체 변수이름에 메모리를 동적 할당하여 필요한 구조체 포인터 배열을 만들 수 있다. 등록시 필요한 구조체 메모리를 동적할당하고 그 이후 내부변수(이름, 전화번호, 생일등)들을 최대 100bytes(널 문자 제외 크기) 안에서 필요한 크기만큼만 메모리를 할당 받아 사용한다. 아래 코드 및 12장 동적할당 강의자료 예제4와 예제3을 참조한다.

```
printf("Max_num:"); // 고정
scanf("%d", &max_num);

//max_num을 이용하여 구조체 변수 이름으로 구조체 포인터 포인터 배열로 메모리 동적 할당
//새로운 연락처가 등록될 때 구조체 메모리 할당
//구조체 내부 멤버들은 입력받은 길이+1만 메모리 할당
//삭제시 내부 멤버 정리 및 구조체 메모리 정리
```

새로운 사람의 정보를 등록할 때마다 예제 4와 3처럼 구조체 메모리를 할당하고, 구조체 내부 멤버들에 대한 메모리를 할당 받아 입력하고, 자료가 삭제될 때 메모리를 free하는 것을 잊지 말아야 한다.

#### ◎ 지원 기능

구조체 포인터 배열을 사용하여 1단계와 동일한 기능을 구현한다.

#### ◎ 유저 인터페이스

1단계와 동일하다. 단, 최초 메뉴 출력 이전에 최대 입력가능한 정보의 개수를 입력받는데 아래 코드를 사용한다. (모든 문자를 동일하게 출력해야 oj시스템을 통과한다.)

```
printf("Max_num:");
...
```

### 입력 형식

위에 기술된 명세대로 표준입력(stdin)으로 입력된다. 메뉴 선택 부분에서는 1, 2, 3, 4, 5 중 하나의 숫자가 입력되고, 이름, 전화번호, 생일을 입력하는 부분에서는 각각 하나의 문자열이 입력된다. 잘못된 입력은 없다고 가정한다.

### 출력 형식

위에 기술된 명세대로 표준출력(stdout)으로 출력한다. 안내 화면 및 등록된 사람의 정보를 출력하기 위해서 위에서 제시된 코드를 그대로 사용한다. (공백 하나라도 차이가 나면, 채점 시스템에서 오답으로 처리가 됨을 주의하고, 과제에서 제시된 것 이외의 추가적인 출력문을 사용하지 말 것.)

#### ◎ 아래 2개 포함 20개

#### 입력과 출력의 예 1

아래에서 굵게 밑줄 친 부분은 입력이다.

```
Max_num:
3
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
Name:Hong
Phone_number:01111112222
Birth:20000101
<<1>>
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
Name:KimEunJoo
Phone_number:222111333
Birth:19960303
<<2>>
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
Name:HanGeul
Phone_number:010222333
Birth:19980101
<<3>>
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
OVERFLOW
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:5
```

## 입력과 출력의 예 2

아래에서 굵게 밑줄 친 부분은 입력이다.

```
Max_num:5      // 줄 관련 수정했음
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
Name:KimEunJoo
Phone_number:0001112222
Birth:20001010
<<1>>
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
Name:Park
Phone_number:+820010101013211221321321
Birth:19900101
<<2>>
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:1
Name:LeeEunJoo_MS
Phone_number:3401222
Birth:19950101
<<3>>
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:2
KimEunJoo 0001112222 2001010
LeeEunJoo_MS 3401222 19950101
Park +820010101013211221321321 19900101
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:3
Name:LeeEunJoo_MS
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:2
KimEunJoo 0001112222 2001010
Park +820010101013211221321321 19900101
*****Menu*****
<1.Registration><2.ShowAll><3.Delete><4.FindByBirth><5.Exit>
Enter_the_menu_number:5
```

## 제출기한 및 방법

- \* OJ시스템에 20XX년 XX월 XX일 XX:XX까지 제출
- \* 프로그램 소스에는 반드시 적절한 들여쓰기 (Indentation)를 하고 주석을 달 것.
- \* 표준 함수만을 이용하여 프로그램을 작성할 것 (algorithm 헤더와 STL 사용 불가)
- \* 보고서 등 기타 제출물 없음