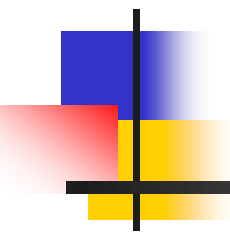


C언어 (CSE2035)

(Chap11. Derived types-enumerated, structure, and union) (2-2)



Ji-Hwan Kim, Ph.D.

Dept. of Computer Science and Engineering

Sogang University

Seoul, Korea

Tel: +82-2-705-8924

Email : kimjihwan@sogang.ac.kr



제출 형식

1. 각 문제에 대한 소스 코드를 압축하여 사이버캠퍼스에 업로드
 - 압축 파일명: "[실습#]학번_이름.zip" (#은 실습번호)
 - 각 소스코드 파일명: "cp실습번호_학번_p문제번호.c"
1. COPY 등의 문제 발생 시 실습 0점 및 각종 불이익을 줄 것



Practice 1

■ 한 회사의 직원들의 정보를 저장할 수 있는 구조체를 다음과 같이 저장한다.

```
Int employee_id; // 직원 ID
char name[40]; // 직원 이름
enum Gender gender; // 성별 (Male = 0, Female = 1)을 가지는 enum
dept department; // 직원이 속한 부서에 대한 struct이다.
double salary; // 직원의 월급
```

■ 제한 조건

- 결과는 **main** 함수 내에서 출력하도록 한다.
- 함수의 원형은 다음과 같다.
- 전역변수 사용 금지

```
typedef struct {
    char dept_name[25];
    int dept_no;
}dept;

enum Gender { Male = 0, Female };

typedef struct{
    int employee_id;
    char name[25];
    enum Gender gender;
    dept department;
    double salary;
}employee_data;

int search(employee_data *e, int num, int keyword);
employee_data update(employee_data e, int n);
```



Practice 1

■ 프로그램이 시작되면 **input file**로부터 정보들을 파일 포인터를 이용하여 읽어온다.

※ **Input file : 16_input.txt**

■ **Input file** 내에는 **n**명에 대한 **id / name / gender / department / department_no / salary** 순으로 저장되어 있다.

```
26
20151053 / Natalie Lewis / Male / FINANCE / 01 / 3000
20161918 / Chloe Christian / Male / SALES / 02 / 3540
20161317 / Emolys Evans / Male / SALES / 02 / 2630
...
```

■ **Input file**에서 읽어 들인 내용으로, 직원의 정보에 대한 예는 다음과 같다.

```
employee_data e1 = {20151053, "Natalie Lewis", Male, {FINANCE, 1}, 3000}
```

■ 직원들의 정보를 이용해서 해당 직원이 존재하는지 검색하고 해당 직원의 정보를 업데이트 할 수 있도록 한다. 직원의 정보를 업데이트 할 때는 **department_no**를 활용하여 부서명과 해당 부서 번호를 변경한다.



Practice 1

■ 부서번호와 부서명은 다음과 같다

{FINANCE, 1}, {SALES, 2}, {RESEARCH, 3}, {PLANNING, 4}

■ 프로그램 실행 예시

```
gr120160221@csp1:~/exercise/17$ ./a.out
Employee ID : 20161403
-----Employee Data-----
20161403 / Vincent Newton / Male / RESEARCH / 3 / 2400.00
-----Update Information-----
Input the department Number : 1
-----Updated employee data-----
20161403 / Vincent Newton / Male / FINANCE / 1 / 2400.00
Continue? (y/n) : y
Employee ID : 20141582
-----Employee Data-----
20141582 / MeanSon King / Female / PLANNING / 4 / 8000.00
-----Update Information-----
Input the department Number : 4
-----Updated employee data-----
20141582 / MeanSon King / Female / PLANNING / 4 / 8000.00
Continue? (y/n) : y
Employee ID : 20151918
No employee
Continue? (y/n) : n
```



Practice 2

로그인 프로그램을 만들어보자.

프로그램이 시작되면 “customer.txt” 파일로부터 회원 정보를 읽고, 회원 정보와 일치되는 사용자가 로그인을 할 경우 “로그인 성공”을, 아닌 경우에는 경고창을 띄어 로그인 프로그램을 만들어보자.

다음과 같은 구조체를 구성하고 회원 정보를 저장한다.

```
typedef struct
{
    int number;
    char id[50];
    char pw[50];
    char name[50];
    char address[200];
}customer;
```



Practice 2

본 과제에서 사용하는 “customer.txt” 파일에는 다음과 같이 정보가 저장되어 있다.
줄 바꿈으로 각 행을 나타내며, 쉼표를 통해 각 행의 열을 구분한다.

```
(회원번호,아이디,비밀번호,이름,주소)
1,ididid,pwpwpw,김서강,서울특별시 마포구 백범로35
2,sment,sunny,이수만,서울특별시 강남구 삼성로648
3,jypent,twice,박진영,서울특별시 강동구 강동대로205
4,ygent,blackpink,양현석,서울특별시 마포구 희우정로1길3
5,kda,idle,riot,서울특별시 강남구 삼성1동 테헤란로521
.....
```

파일의 첫 행에는 파일의 열이 의미하는 정보가 저장되어 있다. 그 다음 행부터는 사용자의 정보가 들어있으며, 이는 순서대로 회원번호, 아이디, 비밀번호, 이름, 주소가 각각 들어있다.

Practice 2

창이 시작되면 **id**와 **pw**를 입력 받아 로그인을 진행한다.

만약 일치하는 회원이 있다면 회원의 정보를 출력하고 프로그램을 종료한다.

만약 일치하는 회원이 없다면 다시 로그인 창으로 들어가 로그인을 진행한다. 만약 3회 이상 일치하지 않는 정보가 들어온다면 프로그램을 종료한다.

Example

```
hinata56@dbpro:~/cprog/prac/16$ ./a.out
로그인 프로그램
아이디 : kda
패스워드 : idle
=====
회원번호 : 5
아이디 : kda
패스워드 : idle
이름 : riot
주소 : 서울특별시 강남구 삼성1동 테헤란로 521
```

```
hinata56@dbpro:~/cprog/prac/16$ ./a.out
로그인 프로그램
아이디 : hearthstone
패스워드 : blizzard
일치하는 회원이 없습니다.
아이디 : Fortnite
패스워드 : epicgames
일치하는 회원이 없습니다.
아이디 : LostArk
패스워드 : smilegate
일치하는 회원이 없습니다.
```




Practice 3

포켓몬 마스터들의 정보를 입력받아 데이터베이스를 구축하는 프로그램을 작성하자.
‘user.txt’에는 포켓몬 마스터들의 정보가 저장되어 있다.

‘user.txt’는 다음과 같은 구조로 이루어진다.

```
Red kanto 25 Pikachu 81 Electric Charm,QuickAttack,Thunderbolt,Thunder
Lance Johto 130 Gyarados 44 Water,Flying RainDance,Surf,Flail,HyperBeam
Green kanto 65 Alakazam 54 Psychic Disable,Recover,Psychic,Reflect
....
```

각 행들은 줄바꿈으로 구분되며, 각 열은 탭(‘\t’)으로 구분된다.

- 1) ‘Red’: 포켓몬 마스터의 이름
- 2) ‘kanto’: 포켓몬 마스터의 출신 지역
- 3) ‘25’: 포켓몬 마스터가 데리고 다니는 포켓몬의 도감 번호
- 4) ‘Pikachu’: 포켓몬 마스터가 데리고 다니는 포켓몬의 이름 (한 명 당 최대 6마리를 데리고 다닐 수 있다!)
- 5) ‘81’: 포켓몬 마스터가 데리고 다니는 포켓몬의 레벨
- 6) ‘Electric’: 포켓몬 마스터가 데리고 다니는 포켓몬의 타입 (하나의 포켓몬은 최대 2개의 타입을 가진다.)
- 7) ‘Charm,QuickAttack,Thunderbolt,Thunder’: 포켓몬 마스터가 데리고 다니는 포켓몬의 기술명. (하나의 포켓몬은 반드시 4개의 기술을 가지며, 기술명은 공백을 포함하지 않고 쉼표 ‘,’로 구분된다.)

Practice 3

다음과 같은 구조체를 사용한다.

```
typedef struct {
    int num;
    char name[20];
    int level;
    char type1[30];
    char type2[30];

    char sk1[20];
    char sk2[20];
    char sk3[20];
    char sk4[20];
}dex;

typedef struct {
    int inx;
    char name[10];
    char region[10];
    dex poke[6];
}
user;
```

‘user’ type의 구조는 다음과 같다.

int inx	현재 user가 가지고 있는 포켓몬의 수
char name[10]	User의 이름
char region[10]	User의 출신지역
dex poke[6]	User가 가지고 있는 포켓몬의 정보.

‘dex’ type의 구조는 다음과 같다.

int num	포켓몬의 도감 번호
char name[10]	포켓몬의 이름
int level	포켓몬의 레벨
char type1,2[30]	포켓몬의 타입
Char sk1~4[30]	포켓몬의 기술명.

Type user

레드
(kanto)

Type dex

25
피카츄 (lv 81)
...

196
에브리 (lv 73)
...

2
이상해꽃 (lv 77)
...

143
잠만보 (lv 75)
...

6
리자몽 (lv 77)
...

9
거북왕 (lv 77)
...

Practice 3

입력 : 'user.txt'의 파일을 통해 입력 받는다.

출력 :

첫 번째 줄에 user.txt에 존재하는 포켓몬 마스터의 수를 출력한다.

다음 줄 부터 각 포켓몬 마스터의 이름과 데리고 다니는 포켓몬들의 모든 정보들을 출력한다.

ex) [Name] Pikachu(25)

= 데리고 있는 포켓몬의 도감번호가 25번이고, Pikachu라는 이름을 가진다.

```
hinata56@dbpro:~/cprog/prac/16$ ./a.out
Number of Users = 3
```

```
[Name] Red(kanto)
[Pokemon]
[Name]Pikachu(25)
[Level]81
[Type]Electric
[Skill1]Charm
[Skill2]QuickAttack
[Skill3]Thunderbolt
[Skill4]Thunder
```

```
[Name]Espeon(196)
[Level]73
[Type]Psychic
[Skill1]Mud-Slap
[Skill2]Swift
[Skill3]Reflect
[Skill4]Psychic
```

```
[Name]Venusaur(2)
[Level]77
[Type1]Grass
[Type2]Poison
[Skill1]SunnyDay
[Skill2]GigaDrain
[Skill3]Synthesis
[Skill4]SolarBeam
```

```
[Name]Snorlax(143)
[Level]75
[Type]Normal
[Skill1]Amnesia
[Skill2]Snore
[Skill3]Rest
[Skill4]BodySlam
```

```
[Name] Lance(Johto)
[Pokemon]
[Name]Gyarados(130)
[Level]44
[Type1]Water
[Type2]Flying
[Skill1]RainDance
[Skill2]Surf
[Skill3]Flail
[Skill4]HyperBeam
```

```
[Name]Charizard(6)
[Level]46
[Type1]Fire
[Type2]Flying
[Skill1]Flamethrower
[Skill2]WingAttack
[Skill3]Slash
[Skill4]HyperBeam
```

```
[Name]Dragonite(149)
[Level]47
[Type1]Dragon
[Type2]Flying
[Skill1]Twister
[Skill2]ThunderWave
[Skill3]Blizzard
[Skill4]HyperBeam
```

```
[Name]Aerodactyl(143)
[Level]46
[Type1]Rock
[Type2]Flying
[Skill1]RockSlide
[Skill2]AncientPower
[Skill3]WingAttack
[Skill4]HyperBeam
```

```
[Name]Dragonite(149)
[Level]47
```