

15강

C_PROGRAMMING



구조체란

❖ 정의

- 여러 자료형으로 메모리의 연속된 공간에 할당 받아 사용하는 사용자 정의 자료형

❖ 기본 구조

```
struct tag_name{  
    type    variable_name;  
    type    variable_name;  
    type    variable_name;  
    .  
    .  
};
```

}

모든 문장에 마지막은
항상 붙여야 하죠^^

구조체 분석

```
struct tag_name{  
    type variable_name;  
    type variable_name;  
    type variable_name;  
    .  
    .  
    .  
};
```

- ❖ struct
 - 구조체를 정의하기 위해 사용되는 자료형
- ❖ tag_name
 - 구조체를 구분하기 위해 사용되는 이름
- ❖ type
 - 각 원소들의 자료형
- ❖ variable_name
 - 구조체 원소, 필드, 멤버, 구성요소...

구조체 선언

```
#include <stdio.h>
struct Student{
    int id;
    char name[12];
    int kor, eng, math;
    char address[50];
};
```

```
int main()
{
    struct Student ST = { 1001, "hacker", 90, 80, 60, "서울시 동대문구" };
    printf("ST의 ID : %d \n", ST.id);
    printf("ST의 이름 : %s \n", ST.name);
    printf("ST의 성적(국, 영, 수) : %d , %d , %d\n", ST.kor, ST.eng, ST.math);
    printf("ST의 주소 : %s \n", ST.address);
    return 0;
}
```

서울시	address
60	math
80	eng
90	kor
hacker	name
1001	ID

구조체 선언

❖ 핸드폰 기종, 가격, 전화번호

```
#include <stdio.h>
#include <string.h>
typedef struct handphone{
    char name[20];
    int  price;
    char phone_num[14];
}HANDPHONE;
main()
{
    HANDPHONE phone;
    strcpy(phone.name, "iPhone5s");
    phone.price = 800000;
    strcpy(phone.phone_num, "010-8505-6235");

    printf("%s\n", phone.name);
    printf("%d\n", phone.price);
    printf("%s\n", phone.phone_num);
}
```

```
gets(phone.name);
scanf("%d", &phone.price);
scanf("%s", phone.phone_num);
```

구조체 멤버 참조

❖ 선언하며 변수 생성

```
typedef struct handphone{  
    char name[20];  
    int  price;  
    char phone_num[14];  
}HANDPHONE;
```

```
main()
```

```
{
```

```
    HANDPHONE phone;
```

```
    strcpy(phone.name, "INFISCAP");
```

```
    phone.price = 800000;
```

```
    strcpy(phone.phone_num, "010-8505-6235");
```

```
}
```

여러 구조체 변수

```
#include <stdio.h>
#include <string.h>
typedef struct handphone{
    char name[20];
    int  price;
    char phone_num[14];
}HANDPHONE;
main()
{
    HANDPHONE phone1, phone2;
    strcpy(phone1.name, "INFISCAP");
    phone1.price = 800000;
    strcpy(phone1.phone_num, "010-8505-6235");
    phone2 = phone1;
    phone2.price = 900000;
}
```

실습문제 1

❖ 구조체 멤버에 값 넣기

- 사용자로부터 ID,이름,성적,주소를 입력 받은 후 아래와 같이 출력되게 하라.

<< ST 2 의 구조체의 정보 >>

ST2의 ID : 1002

ST2의 이름 : hackers

ST2의 성적(국,영,수) : 95, 93, 77

ST2의 주소 : 서울시 강남구 신사동

ST2의 합계 : 265

ST2의 평균 : 88.33

ST2의 등급 : 우

구조체의 크기

- ❖ 구조체의 내용중에서 가장 큰 자료형을 기본 단위로 설정하고 word alignment 단위로 메모리 할당
- ❖ word alignment
 - 각 자료형의 크기별로 배수로 메모리에 저장
- ❖ 자료형별 메모리 할당 위치

자료형	크기	할당 위치
char	1byte	자유로이 메모리 주소 할당
short int	2byte	2의 배수로 메모리 주소 할당
int, float	4byte	4의 배수로 메모리 주소 할당
double	8byte	8의 배수로 메모리 주소 할당

예제

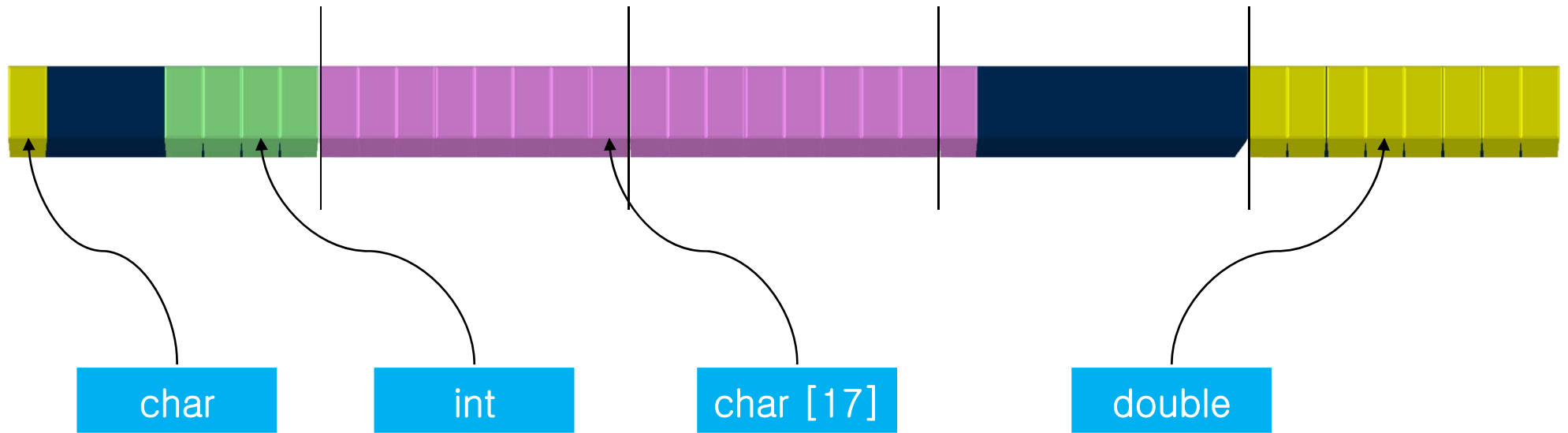
```
#include <stdio.h>
```

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

```
typedef struct test{  
    char    ch;           //1byte  
    int     num;          //4byte  
    char    str[17];      //17byte  
    double  double_num;  //8byte  
}TEST;
```

```
main()  
{  
    TEST s_test;  
    printf("struct size: %d\n", sizeof(s_test)); //결과는 30byte??  
}
```

결과



문자열

❖ 문자열을 비교하는 함수

```
int strcmp(char * dest, const char * src);
```

- string compare.
- 문자열을 비교하여 어떠한 결과를 리턴한다.
- 두 문자열이 같을 경우 0을 반환 그렇지 않을 경우 음수나 양수를 반환

❖ 문자열을 정해진 크기만큼만 비교하는 함수

```
int strncmp(char * dest, const char * src , size_t n);
```

- 비교하는 문자열의 최대길이는 n을 넘지 않는다.
- 정해진 구역까지만 비교 할 수 있어 유용하게 사용 가능하다.

실습예제

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

```
int main()
```

```
{
```

```
    char ar1[10] = "Apple";
```

```
    char ar2[10] = "Apple";
```

```
    int tmp;
```

```
    tmp = strcmp(ar1, ar2);
```

//두 문자열이 같을 때 0을 반환

```
    printf("비교 값 %d\n", tmp);
```

```
    tmp = strcmp(ar1, "Orange");
```

//뒤의 문자열의 값이 더 클 때 -1 반환

```
    printf("비교 값 %d\n", tmp);
```

```
    tmp = strcmp(ar1, "Appl d");
```

//앞의 문자열의 값이 더 클 때 1을 반환

```
    printf("비교 값 %d\n", tmp);
```

```
    tmp = strncmp(ar1, "Apppp", 3);
```

```
    printf("비교 값 %d\n", tmp);
```

//앞의 3문자만 비교하여 결과를 반환

```
    return 0;
```

```
}
```

실습 문제 2

● 사용자 인증 톨 만들기

1. 사용자 정보(ID, Password)를 저장할 수 있는 구조체를 만들고, 사용자로부터 ID와 Password를 입력 받은 후 인증을 처리하는 톨을 만들어라. (사용자 정보는 미리 입력되어 있는 상태에서 시작하고, 문자열 조작 함수를 이용해서 구현할 것.)

2. 예시

사용자 인증이 필요합니다. ID와 Password를 입력하세요.

ID: user1

Password: test

틀렸습니다. ID와 Password를 다시 입력하세요.

ID: user1

Password: password123

인증을 통과했습니다.

감사합니다.

구조체 배열

❖ 정의

- 구조체를 배열을 이용하여 여러 데이터를 처리

❖ 예제

```
typedef struct member{
    char name[20];
    char sex[6];
    int age;
    char phone_num[14];
}MEMBER;
main()
{
    MEMBER man[5]={ "Lee sun_sim", "man", 37, "010-xxx-yyyy",
                    "Hong gil_dong", "man", 16, "-",
                    "Yoo kwan_sun", "women", 15, "011-xxx-yyyy",
                    "Kang kam_chan", "man", 27, "016-xxx-yyyy",
                    "Sin saimdang", "women", 42, "019-xxx-yyyy"};
}
```

man[1].name man[0] man[0].age

	name	sex	age	Phone_num
man[0]	Lee sun_sin	man	37	010-xxx-yyyy
man[1]	Hong gil_dong	man	16	-
man[2]	Yoo kwan_sun	women	15	011-xxx-yyyy
man[3]	Kang kam_chan	man	27	016-xxx-yyyy
man[4]	Sin saimdang	women	42	019-xxx-yyyy

실습 문제 3

● 사용자 인증 툴 만들기

1. 사용자 정보(ID, Password)는 구조체를 이용해 저장하고, 로그인, 회원 가입, 회원 탈퇴 기능을 수행해야 한다. (회원의 수는 최대 100명)

2. 예시

원하는 기능을 선택하세요.

1. 로그인
2. 회원 가입
3. 회원 탈퇴
4. 종료

Select...

Quiz

- ❖ Man이라는 구조체를 만드시오
- ❖ Computer라는 구조체를 만드시오
- ❖ Student라는 구조체를 만드시오
- ❖ Monitor라는 구조체를 만드시오
- ❖ 위의 구조체를 이용하여 데이터를 입력받고 출력하는 프로그램을 작성하시오

Quiz

- ❖ 자동차 영업사원 관리 프로그램 작성
- ❖ 구조체 멤버는 최소 3개이상 만들며 매출액은 필히 추가할것
- ❖ 구조체 배열로 최소 100개 이상 지정할것
- ❖ 매출액에 따른 오름차순으로 데이터를 정렬 시킬것
- ❖ 최소매출 사원과 최대 매출 사원을 표기할것
- ❖ 각 차종별 판매 대수를 분포도를 이용하여 표기할것

구조체 포인터

❖ -> 연산자를 이용한 표현 방법

```
typedef struct handphone{
    char name[20];
    int  price;
    char phone_num[14];
}HANDPHONE, *PHANDPHONE;
main()
{
    HANDPHONE      phone;
    PHANDPHONE      * ptrPhone;
    ptrPhone =& phone;
    ptrPhone ->price=750000;
}
```

예제

```
typedef struct handphone{
    char name[20];
    int  price;
    char phone_num[14];
}HANDPHONE, *PHANDPHONE;
main(){
    HANDPHONE      phone;
    PHANDPHONE      ptrPhone;

    ptrPhone = &phone;
    strcpy(phone.name, "INFISCAP");
    phone.price = 800000;
    strcpy(phone.phone_num, "010-8505-6235");
    ptrPhone->price = 900000
}
```

예제

```
typedef struct handphone{
    char name[20];
    int  price;
    char phone_num[14];
}HANDPHONE, *PHANDPHONE;
main(){
    HANDPHONE      phone, phone2;
    PHANDPHONE      ptrPhone;

    ptrPhone = &phone;
    strcpy(phone.name, "INFISCAP");
    phone.price = 800000;
    strcpy(phone.phone_num, "010-8505-6235");
    phone2 = *ptrPhone;
}
```

Bit 필드

❖ 구조체의 원소를 bit단위로 사용

❖ 형식

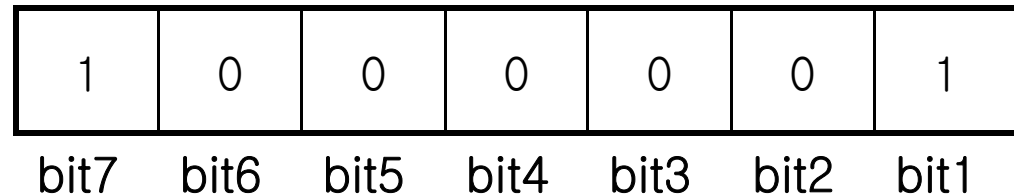
```
struct tag_name {  
    unsigned member1 : bit_size;  
    unsigned member2 : bit_size;  
    ....  
};
```

❖ type은 int 또는 unsigned

예제

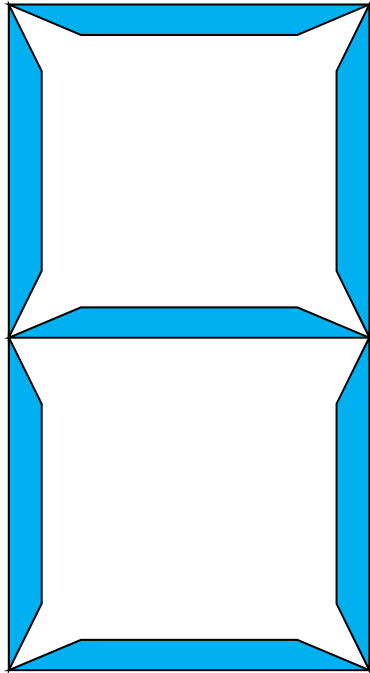
```
typedef struct ASCII_Code{  
    unsigned    bit1 : 1;  
    unsigned    bit2 : 1;  
    unsigned    bit3 : 1;  
    unsigned    bit4 : 1;  
    unsigned    bit5 : 1;  
    unsigned    bit6 : 1;  
    unsigned    bit7 : 1;  
}ASCII_CODE, *PASCII_CODE;
```

```
main()  
{  
    ASCII_CODE  ch;  
  
    ch.bit1=1;  
    ch.bit7=1;  
    printf("%c", ch);  
}
```



Quiz

❖ 전자시계 만들기



구조체와 함수

```
typedef struct handphone{
    char name[20];
    int price;
    char phone_num[14];
}HANDPHONE, *PHANDPHONE;

void PriceCheck(int price);
void main(){
    HANDPHONE phone;

    phone.price=750000;
    PriceCheck(phone.price);
}

void PriceCheck(int price){
    if(price>700000)
        printf("70만원에 주세요\n");
}
```

구조체 인자

```
typedef struct handphone{
    char name[20];
    int price;
    char phone_num[14];
}HANDPHONE, *PHANDPHONE;

void PriceCheck(HANDPHONE phone);
void main(){
    HANDPHONE phone;

    phone.price=750000;
    PriceCheck(phone);
}
void PriceCheck(HANDPHONE phone){
    if(phone.price>700000)
        printf("70만원에 주세요\n");
}
```

구조체 포인터 인자

```
typedef struct handphone{
    char name[20];
    int  price;
    char phone_num[14];
}HANDPHONE, *PHANDPHONE;

void PriceCheck(PHANDPHONE phone);
void main(){
    HANDPHONE  phone;

    phone.price=750000;
    PriceCheck(&phone);
}
void PriceCheck(PHANDPHONE phone){
    phone->price =700000
}
```

구조체 참조 구조체

```
typedef struct{  
    char        com[20];  
    int         size;  
    int         ea;  
}memory;
```

```
typedef struct{  
    memory      ddr;  
    cpu         _cpu;  
}_computer;
```

```
_computer      computer;  
computer.ddr.size=256;
```

자기 참조 구조체

- ❖ 구조체는 자기 자신을 참조할 수 없다.
- ❖ 자기 참조 구조체(self-referential structure)
 - 자기 자신을 가리키는 포인터를 멤버로 가지는 구조체
 - 리스트(list)나 트리(tree)와 같은 자료 구조 구현에 사용

```
struct book {                /* 자기 참조 구조체 */  
    char title[80];  
    char author[80];  
    struct book *nextbook;  
};
```



공용체

❖ 정의

- 여러 자료형들을 하나의 공간에 위치 하여 메모리를 공유하는 것

❖ 장.단점

- 공간 절약
- 가장 큰 멤버의 크기로 공간 확보
- 메모리의 데이터가 쉽게 변화할 수 있다.
- 한개 이상의 멤버 사용 불가

❖ 사용예

```
union test{  
    char    ch;  
    int     num;  
    double  flt;  
    char    str[10];  
}man;
```



<man>

Quiz

❖ 매장 관리 프로그램

❖ 기본 메뉴

- 1. 상품 선택
- 2. 판매 수량
- 3. 반품 수량
- 4. 최고 매출 상품
- 5. 최저 매출 상품
- 6. 최고 반품 상품
- 7. 총 매출액
- 8. 종료

❖ 구상