

C Programming

복합 데이터 유형 (Composite Data Types)



Seo, Doo-Ok

Clickseo.com

clickseo@gmail.com



목 차



백문이불여일타(百聞而不如一打)

- 구조체의 이해
- 구조체 배열과 포인터.함수
- 공용체와 열거형



구조체



- 구조체의 이해

백문이불여일타(百聞而不如一打)

- 중첩 구조체

- 비트 필드 구조체

- 구조체 배열과 포인터.함수

- 공용체와 열거형



구조체의 이해 (1/9)

- **구조체**(Structure)

- 서로 연관된 원소들의 집합을 하나의 이름으로 묶어 놓은 것
 - 변수들의 집합체(단, 변수들은 서로 다른 자료형을 가질 수도 있다)
- 배열과 구조체의 차이
 - 구조체에 있는 원소들은 동일한 자료형 이거나 다른 자료형 일 수 있다.

name	kor	eng	math	tot	ave
------	-----	-----	------	-----	-----

temp

구조체의 이해 (2/9)

- 구조체 형식 정의

1. 구조체 변수(Structure Variable)
2. 태그 된 구조체(Tagged Structure)
3. 형식 정의된 구조체(Type-Defined Structure)

```
// Structure Variable
```

```
struct {  
    ...  
} variableIdentifier;
```

```
// Tagged Structure
```

```
struct TagName {  
    ...  
} variable_identifier;  
  
struct TagName variableIdentifier;
```

```
// Type-Defined Structure
```

```
typedef struct {  
    ...  
} NewType;  
  
NewType variableIdentifier;
```

구조체의 이해 (3/9)

- 구조체 형식 정의: 구조체 변수

- 구조체 변수

```
// Structure Variable  
struct {  
    ... fieldList ...  
} variableIdentifier;
```

```
// 구조체 형식 정의 및 변수 선언  
struct {  
    char    name[12];  
    int     kor, eng, math, tot;  
    float   ave;  
} temp;
```

```
// 구조체 형식 정의 및 변수 선언  
struct {  
    char    name[12];  
    int     kor, eng, math, tot;  
    float   ave;  
} a, b, c;
```

구조체의 이해 (4/9)

- 구조체 형식 정의: 태그 된 구조체

- 태그 된 구조체

- 구조체에 태그명을 부여함으로써, 태그명을 사용하여 변수나 매개변수, 반환 값을 정의할 수 있다.

```
// 구조체 형식 정의
struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
};
```

```
// 구조체 변수 선언
struct _score    temp;
// _score        temp;    // error C2065: '_score': 선언되지 않은 식별자입니다.
// temp          temp;    // error C2065: 'temp': 선언되지 않은 식별자입니다.
```

```
void    OUTPUT(struct _score temp);    // 함수 호출
```

```
// Tagged Structure
struct TagName {
    ...
} variableIdentifier;

struct TagName variableIdentifier;
```

구조체의 이해 (5/9)

- 구조체 형식 정의: 형식 정의된 구조체

- 형식 정의된 구조체

- 태그형 구조체과 다른 점
 - typedef 키워드를 선언부 앞에 추가하는 것
 - 블록문의 끝에 있는 식별자가 변수가 아닌 형식 정의 이름

```
// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;
```

```
// 구조체 변수 선언
// struct _score    temp;
SCORE                temp;
```

```
void    OUTPUT(SCORE temp);    // 함수 호출
```

```
// Type-Defined Structure
```

```
typedef struct {
```

```
    ...
```

```
} NewType;
```

```
NewType variableIdentifier;
```

구조체의 이해 (6/9)

- 구조체형 변수: 선언 및 초기화

- 초기값은 중괄호를 사용하며 콤마로 분리

- 단, 구조체에 정의된 각 구조체 멤버들의 형식과 일치해야 한다.

```
// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

// 구조체 변수 선언 및 초기화
SCORE temp = {"홍길동", 70, 80, 90, 240, 80.0};
```

temp

"홍길동"	70	80	90	240	80.0
-------	----	----	----	-----	------

구조체의 이해 (7/9)

- 구조체형 변수: 멤버 접근과 크기

- 구조체 멤버 접근

- 개별 필드 참조: **구두점 표기법**(Dot Notation)
- 구조체 변수명과 멤버 이름 사이를 구두점 연산자(.)로 구분하여 사용

temp	" 홍길동 "	70	80	90	240	80.0
	temp.name	temp.kor	temp.eng	temp.math	temp.tot	temp.ave

- 구조체의 크기: 각 멤버들이 차지하는 메모리 용량의 총합

`sizeof (struct _score) = 32 Bytes`

`sizeof (SCORE) = 32 Bytes`

구조체의 이해 (8/9)

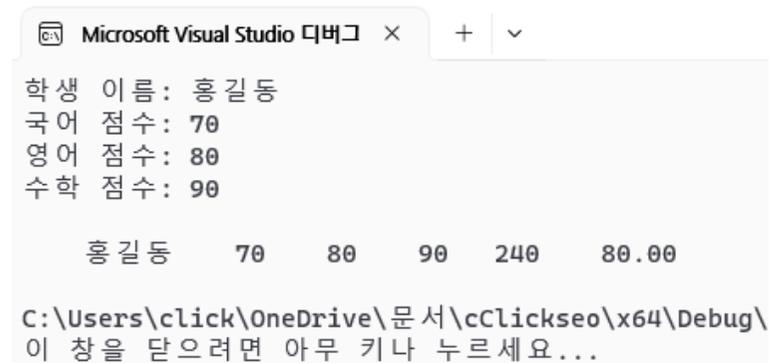
예제 9-1: 구조체 형식 정의 -- 변수 선언과 멤버 접근

(1/2)

```
#include <stdio.h>

// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

int main(void)
{
    // 구조체 변수 선언
    SCORE temp;
```



```
Microsoft Visual Studio 디버그 x + v
학생 이름: 홍길동
국어 점수: 70
영어 점수: 80
수학 점수: 90

홍길동 70 80 90 240 80.00
C:\Users\click\OneDrive\문서\cClickseo\x64\Debug\
이 창을 닫으려면 아무 키나 누르세요...
```

temp	" 홍길동"	70	80	90	240	80.0
	temp.name	temp.kor	temp.eng	temp.math	temp.tot	temp.ave

구조체의 이해 (8/9)

예제 9-1: 구조체 형식 정의 -- 변수 선언과 멤버 접근

(2/2)

```
// 한 학생 성적 입력
printf("학생 이름: ");
gets_s(temp.name, sizeof(temp.name)); // gets(temp.name);
printf("국어 점수: ");
scanf_s("%d", &temp.kor); // scanf("%d", &temp.kor);
printf("영어 점수: ");
scanf_s("%d", &temp.eng); // scanf("%d", &temp.eng);
printf("수학 점수: ");
scanf_s("%d", &temp.math); // scanf("%d", &temp.math);

// 성적 계산: 총점, 평균
temp.tot = temp.kor + temp.eng + temp.math;
temp.ave = (float)temp.tot / 3;

// 한 학생의 성적 출력
printf("\n%10s %5d %5d %5d %5d %8.2f \n",
      temp.name, temp.kor, temp.eng, temp.math, temp.tot, temp.ave);

return 0;
}
```

구조체의 이해 (9/9)

예제 9-2: 구조체 형식 정의 -- Padding Bytes

```
#include <stdio.h>

typedef struct {
    char   ch;
    int    i;
    double d;
} TYPE1;

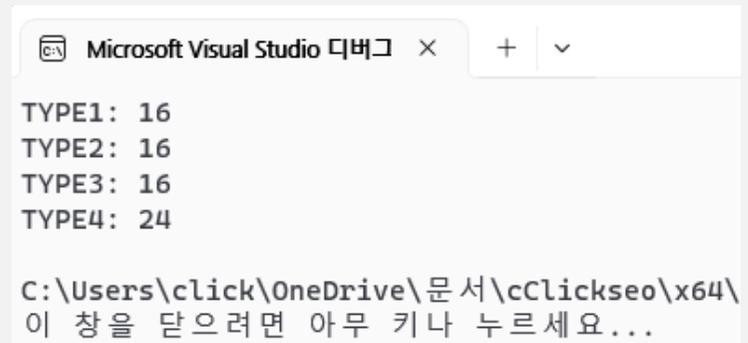
typedef struct {
    int    i;
    char   ch;
    double d;
} TYPE2;

typedef struct {
    double d;
    char   ch;
    int    i;
} TYPE3;

typedef struct {
    char   ch;
    double d;
    int    i;
} TYPE4;
```

```
int main(void)
{
    printf("TYPE1: %zd \n", sizeof(TYPE1));
    printf("TYPE2: %zd \n", sizeof(TYPE2));
    printf("TYPE3: %zd \n", sizeof(TYPE3));
    printf("TYPE4: %zd \n", sizeof(TYPE4));

    return 0;
}
```



```
Microsoft Visual Studio 디버그 x + v
TYPE1: 16
TYPE2: 16
TYPE3: 16
TYPE4: 24
C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```



구조체의 이해

중첩 구조체



중첩 구조체 (1/3)

- 중첩 구조체(Nested Struct)

```
// 추천하지 않는 방식
typedef struct {
    struct {
        int year;
        int month;
        int day;
    } Date;
    struct {
        int hour;
        int min;
        int sec;
    } Time;
} TIMESTAMP;
```

```
TIMESTAMP temp;
```

```
// 선호하는 방식
typedef struct {
    int year;
    int month;
    int day;
} DATE;

typedef struct {
    int hour;
    int min;
    int sec;
} TIME;
```

```
typedef struct {
    DATE Date;
    TIME Time;
} TIMESTAMP;
```

```
TIMESTAMP temp;
```

중첩 구조체 (2/3)

예제 9-3: 중첩 구조체 형식 정의 -- 변수 선언과 멤버 접근

(1/2)

```
#include <stdio.h>

// 중첩 구조체 형식 정의
typedef struct {
    int year;
    int month;
    int day;
} DATE;

typedef struct {
    int hour;
    int min;
    int sec;
} TIME;

typedef struct {
    DATE Date;
    TIME Time;
} TIMESTAMP;
```

```
// 추천하지 않는 방식
typedef struct {
    struct {
        int year;
        int month;
        int day;
    } DATE;
    struct {
        int hour;
        int min;
        int sec;
    } TIME;
} TIMESTAMP;

TIMESTAMP temp;
```

중첩 구조체 (3/3)

예제 9-3: 중첩 구조체 형식 정의 -- 변수 선언과 멤버 접근

(2/2)

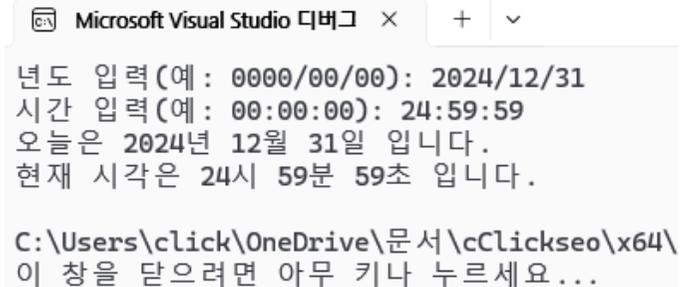
```
int main(void)
{
    // 구조체 변수: 중첩 구조체
    TIMESTAMP temp;

    printf("년도 입력 (예: 0000/00/00): ");
    scanf_s ("%d%c%d%c%d",
              &temp.Date.year, &temp.Date.month, &temp.Date.day );

    printf("시간 입력 (예: 00:00:00): ");
    scanf_s ("%d%c%d%c%d",
              &temp.Time.hour, &temp.Time.min, &temp.Time.sec );

    printf("오늘은 %d년 %d월 %d일 입니다.\n",
           temp.Date.year, temp.Date.month, temp.Date.day );
    printf("현재 시각은 %d시 %d분 %d초 입니다.\n",
           temp.Time.hour, temp.Time.min, temp.Time.sec );

    return 0;
}
```



Microsoft Visual Studio 디버그 × + ▾

년도 입력 (예: 0000/00/00): 2024/12/31
시간 입력 (예: 00:00:00): 24:59:59
오늘은 2024년 12월 31일 입니다.
현재 시각은 24시 59분 59초 입니다.

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...



구조체의 이해

비트 필드 구조체



비트 필드 구조체 (1/4)

- **비트 필드 구조체**(Bit Field Structure)

- 멤버들의 크기가 비트 단위로 나누어져 있는 구조체

```
// 구조체 형식 정의
typedef struct {
    int year;      // 년
    int month;    // 월
    int day;      // 일
} DATE;

// 비트 필드 구조체 형식 정의
typedef struct {
    unsigned int year: 16;    // 0 ~ 9999
    unsigned int month: 4;   // 1 ~ 12
    unsigned int day: 5;     // 1 ~ 31
    unsigned int : 7;        // Reserved           // Padding Bits
} bitFieldDATE;
```

비트 필드 구조체 (2/4)

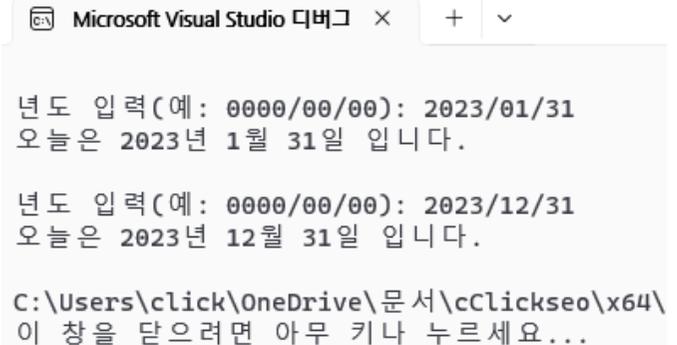
예제 9-4: 비트 필드 구조체 형식 정의 -- 날짜(Date)

(1/2)

```
#include <stdio.h>

// 구조체 형식 정의
typedef struct {
    int year;      // 년
    int month;     // 월
    int day;       // 일
} DATE;

// 비트 필드 구조체 형식 정의
typedef struct {
    unsigned int year: 16; // 0 ~ 9999
    unsigned int month: 4; // 1 ~ 12
    unsigned int day: 5;   // 1 ~ 31
    unsigned int : 7;      // Reserved // Padding Bits
} bitFieldDATE;
```



Microsoft Visual Studio 디버그

년도 입력(예: 0000/00/00): 2023/01/31
오늘은 2023년 1월 31일 입니다.

년도 입력(예: 0000/00/00): 2023/12/31
오늘은 2023년 12월 31일 입니다.

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...

비트 필드 구조체 (3/4)

예제 9-4: 비트 필드 구조체 형식 정의 -- 날짜(Date)

(2/2)

```
int main(void)
{
    DATE          date;

    printf("\n년도 입력(예: 0000/00/00): ");
    scanf_s("%d%c%d%c%d", &date.year, &date.month, &date.day);
    // scanf("%d%c%d%c%d", &date.year, &date.month, &date.day);
    printf("오늘은 %d년 %d월 %d일 입니다.\n", date.year, date.month, date.day);

    // 비트 필드 구조체 변수 선언
    bitFieldDATE  newDate;

    // 비트 필드 구조체 멤버에는 주소 연산자(&) 사용 불가
    // error C2104 : 비트 필드의 '&'이(가) 무시됩니다.
    // scanf_s("%d%c%d%c%d", &newDate.year, &newDate.month, &newDate.day);

    // 임시 변수: 년월일
    int          year, month, day;

    printf("\n년도 입력(예: 0000/00/00): ");
    scanf_s("%d%c%d%c%d", &year, &month, &day);
    // scanf("%d%c%d%c%d", &year, &month, &day);

    newDate.year = year;
    newDate.month = month;
    newDate.day = day;

    printf("오늘은 %d년 %d월 %d일 입니다.\n", newDate.year, newDate.month, newDate.day )
    return 0;
}
```

비트 필드 구조체 (4/4)

예제 9-5: 비트 필드 구조체 형식 정의 -- 논리형(_Bool)

```
#include <stdio.h>
#include <stdbool.h> // bool, true, false

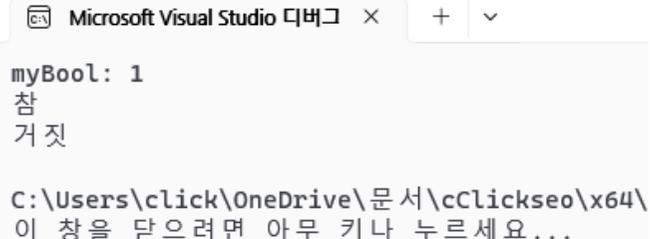
// 비트 필드 구조체 형식 정의
typedef struct _bool {
    unsigned char True: 1; // 1bit
    unsigned char False: 1; // 1bit
    unsigned char : 6; // Reserved // Padding Bits
} _bool;

int main(void)
{
    // _Bool True = 1;
    // _Bool False = 0;
    _bool myBool = { 1, 0 };

    // printf("myBool: %d \n", sizeof(_Bool));
    printf("myBool: %zd \n", sizeof(myBool));

    if(myBool.True == 1) printf("참\n");
    if(myBool.False == 0) printf("거짓\n");

    // error C2070: 'unsigned char' : sizeof 피연산자가 잘못되었습니다.
    // printf("myBool.True: %d \n", sizeof(myBool.True));
    return 0;
}
```



Microsoft Visual Studio 디버그 x + v

```
myBool: 1
참
거짓

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

구조체 배열과 포인터.함수



- 구조체의 이해

백문이불여일타(百聞而不如一打)

- 구조체 배열과 포인터.함수

- 구조체 배열

- 구조체와 포인터

- 구조체와 함수

- 공용체와 열거형



구조체 배열 (1/4)

- 구조체 배열

```
// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

// 구조체 배열 선언
SCORE arr[3];
```

arr[0]	"홍길동"	70	80	90	240	80.0
arr[1]	"이순신"	70	60	80	210	70.0
arr[2]	"이도"	90	80	100	270	90.0

arr

구조체 배열 (2/4)

예제 9-6: 구조체 배열

(1/3)

```
#include <stdio.h>
#define arrMAXSIZE 3

// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

int main(void)
{
    // 구조체 배열
    SCORE arr[arrMAXSIZE];
```

```
Microsoft Visual Studio 디버그 x + v
3 명의 학생 성적을 입력하세요.
1 번째 학생 성적을 입력하세요.
학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 90
2 번째 학생 성적을 입력하세요.
학생 이름 : 이순신
국어 점수 : 70
영어 점수 : 60
수학 점수 : 80
3 번째 학생 성적을 입력하세요.
학생 이름 : 이도
국어 점수 : 90
영어 점수 : 80
수학 점수 : 100

### 성적 결과 ###
1 :   홍길동   70  80  90   240   80.00
2 :   이순신   70  60  80   210   70.00
3 :     이도   90  80 100   270   90.00

C:\Users\click\OneDrive\문서\clickseo\x64\Debug\
이 창을 닫으려면 아무 키나 누르세요...
```

arr[0]	"홍길동"	70	80	90	240	80.0
arr[1]	"이순신"	70	60	80	210	70.0
arr[2]	"이도"	90	80	100	270	90.0

구조체 배열 (3/4)

예제 9-6: 구조체 배열

(2/3)

```
// 여러 명의 학생 성적 입력
printf("%d 명의 학생 성적을 입력하세요. \n", arrMAXSIZE );
for(int i = 0; i < arrMAXSIZE; i++) {
    printf("\n%d 번째 학생 성적을 입력하세요. \n", i + 1);
    printf("학생 이름 : ");
    gets_s(arr[i].name , sizeof(arr[i].name));
    // gets(arr[i].name);

    printf("국어 점수 : ");
    scanf_s("%d", &arr[i].kor );    // scanf("%d", &arr[i].kor );

    printf("영어 점수 : ");
    scanf_s("%d", &arr[i].eng );    // scanf("%d", &arr[i].eng );

    printf("수학 점수 : ");
    scanf_s("%d%c", &arr[i].math );    // scanf("%d%c", &arr[i].math );

    arr[i].tot = arr[i].kor + arr[i].eng + arr[i].math;
    arr[i].ave = (float)arr[i].tot / 3;
}
```

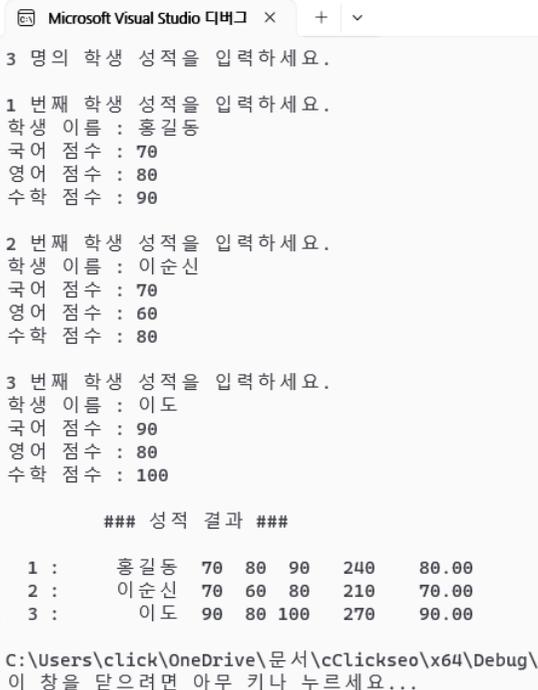
구조체 배열 (4/4)

예제 9-6: 구조체 배열

(3/3)

```
// 여러 명의 학생 성적 결과 출력
printf("\n\t ### 성적 결과 ### \n\n");
for(int i = 0; i < arrMAXSIZE; i++)
    printf("%3d : %10s %3d %3d %3d %5d %8.2f \n",
        i + 1, arr[i].name, arr[i].kor, arr[i].eng, arr[i].math,
        arr[i].tot, arr[i].ave );

return 0;
}
```



Microsoft Visual Studio 디버그

3 명의 학생 성적을 입력하세요.

1 번째 학생 성적을 입력하세요.
학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 90

2 번째 학생 성적을 입력하세요.
학생 이름 : 이순신
국어 점수 : 70
영어 점수 : 60
수학 점수 : 80

3 번째 학생 성적을 입력하세요.
학생 이름 : 이도
국어 점수 : 90
영어 점수 : 80
수학 점수 : 100

성적 결과

1 :	홍길동	70	80	90	240	80.00
2 :	이순신	70	60	80	210	70.00
3 :	이도	90	80	100	270	90.00

C:\Users\click\OneDrive\문서\cClickseo\x64\Debug\
이 창을 닫으려면 아무 키나 누르세요...



구조체 배열과 포인터.함수

구조체와 포인터



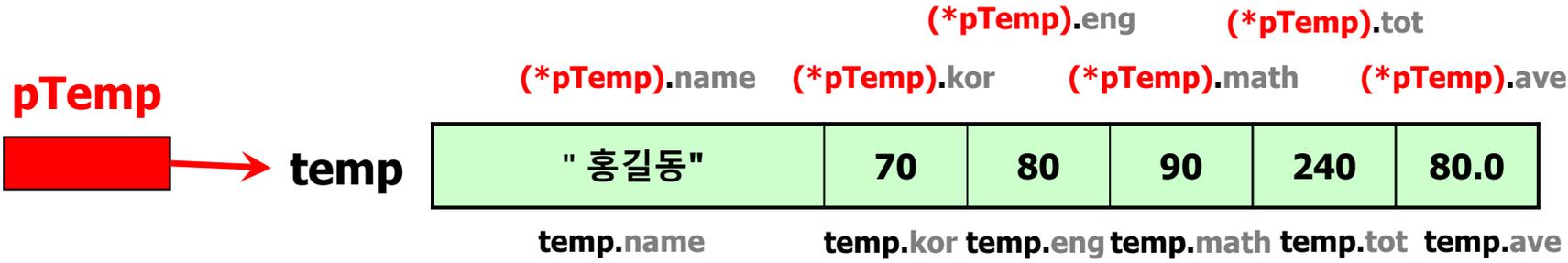
구조체와 포인터 (1/2)

- **간접 표기법**(Indirection Notation)

temp.name **(*pTemp) . name**

```
// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

SCORE    temp;
SCORE    *pTemp = &temp;
```



구조체와 포인터 (2/2)

- **선택 표기법**(Selection Notation)

temp.name

pTemp->name

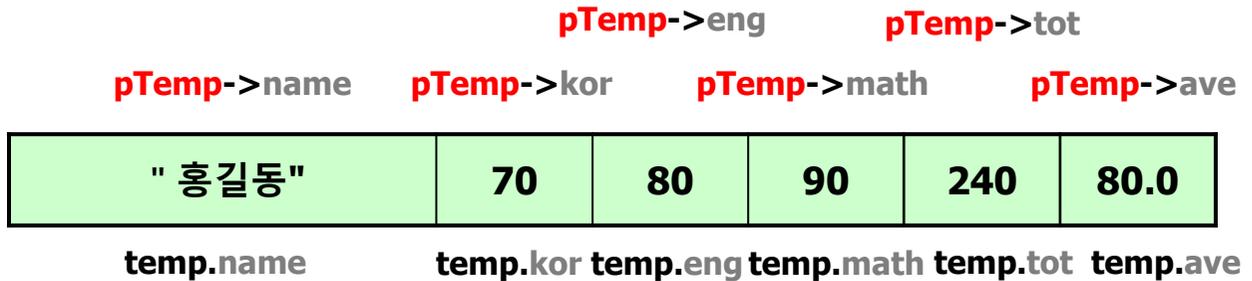
```
// 구조체 형식 정의
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

SCORE    temp;
SCORE    *pTemp = &temp;
```

pTemp



temp





구조체 배열과 포인터, 함수

구조체와 함수



구조체와 함수 (1/10)

- 함수에서 구조체 멤버 접근 방법

1. 구조체 **멤버**를 함수에 전달하고 (pass by value), 함수 내부에서 일반 변수처럼 접근한다.
2. 구조체 **전체**를 함수에 전달하고 (pass by value), 함수 내부에서 각 멤버들을 일반 변수처럼 접근한다.
3. 구조체의 **메모리 주소**를 함수에 전달하고 (pass by address), 함수 내부에서 간접 접근 방법 (간접 표기법 또는 선택 표기법)을 이용하여 구조체 멤버를 접근한다.

구조체와 함수 (2/10)

- 구조체 **멤버**를 함수에 전달

```
temp.tot = SUM(temp.kor, temp.eng, temp.math);
```

int	c	90
int	b	80
int	a	70

SUM 함수
스택 영역

temp

"홍길동"	70	80	90	240	80.0
temp.name	temp.kor	temp.eng	temp.math	temp.tot	temp.ave

main 함수
스택 영역

구조체와 함수 (3/10)

예제 9-7: 구조체 멤버를 함수에 전달 -- pass by value

(1/2)

```
#include <stdio.h>

// 구조체 설계: SCORE(struct _score)
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

// 함수 원형 선언
int    SUM(int a, int b, int c);
float  AVE(int a, int b, int c);

int main(void)
{
    // 구조체 변수 선언
    SCORE s;

    printf("학생 성적을 입력 하세요. \n\n");
    printf("학생 이름 : ");          gets_s(s.name, sizeof(s.name));
    printf("국어 점수 : ");          scanf_s("%d", &s.kor);
    printf("영어 점수 : ");          scanf_s("%d", &s.eng);
    printf("수학 점수 : ");          scanf_s("%d%c", &s.math);
```



Microsoft Visual Studio 디버그 x + v

학생 성적을 입력 하세요...

학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 91

학생 정보

홍길동 70 80 91 241 80.33

구조체와 함수 (4/10)

예제 9-7: 구조체 멤버를 함수에 전달 -- pass by value

(2/2)

```
s.tot = SUM(s.kor, s.eng, s.math);
s.ave = AVE(s.kor, s.eng, s.math);

printf("\n\t ### 학생 정보 ### \n\n");
printf("%10s %3d %3d %3d %5d %8.2f \n",
       s.name, s.kor, s.eng, s.math, s.tot, s.ave);

return 0;
}

// 함수 정의: SUM, AVE
int SUM(int a, int b, int c) {
    return a + b + c;
}

float AVE(int a, int b, int c) {
    return (float)(a + b + c) / 3;
}
```



Microsoft Visual Studio 디버그

학생 성적을 입력 하세요...

학생 이름: 홍길동
국어 점수: 70
영어 점수: 80
수학 점수: 91

학생 정보

홍길동 70 80 91 241 80.33

구조체와 함수 (5/10)

- 구조체 전체를 함수에 전달

```
print(temp);
```

SCORE temp

"홍길동"	70	80	90	240	80.0
temp.name	temp.kor	temp.eng	temp.math	temp.tot	temp.ave

OUTPUT 함수
스택 영역

temp

"홍길동"	70	80	90	240	80.0
temp.name	temp.kor	temp.eng	temp.math	temp.tot	temp.ave

main 함수
스택 영역

구조체와 함수 (6/10)

예제 9-8: 구조체 전체를 함수에 전달 -- pass by value

(1/2)

```
#include <stdio.h>

// 구조체 설계: SCORE(struct _score)
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

// 함수 원형 선언
SCORE input(void);
void print(SCORE);

int main(void)
{
    // 구조체 변수 선언
    SCORE s;

    // 학생 정보 입출력
    s = input();
    print(s);

    return 0;
}
```



Microsoft Visual Studio 디버그

학생 성적을 입력 하세요...

학생 이름: 홍길동
국어 점수: 70
영어 점수: 80
수학 점수: 91

학생 정보

홍길동 70 80 91 241 80.33

구조체와 함수 (7/10)

예제 9-8: 구조체 전체를 함수에 전달 -- pass by value

(2/2)

// 한 명의 학생 정보 입력

```
SCORE input(void) {  
    SCORE temp;
```

```
    printf("학생 성적을 입력 하세요... \n\n");  
    printf("학생 이름: "); gets_s(temp.name , sizeof(temp.name));  
    printf("국어 점수: "); scanf_s("%d", &temp.kor);  
    printf("영어 점수: "); scanf_s("%d", &temp.eng);  
    printf("수학 점수: "); scanf_s("%d", &temp.math);
```

```
    temp.tot = temp.kor + temp.eng + temp.math;  
    temp.ave = (float)temp.tot / 3;
```

```
    return temp;
```

```
}
```

// 한 명의 학생 정보 출력

```
void print(SCORE temp) {
```

```
    printf("\n\t ### 학생 정보 ### \n\n");
```

```
    printf("%10s %3d %3d %3d %5d %8.2f \n",
```

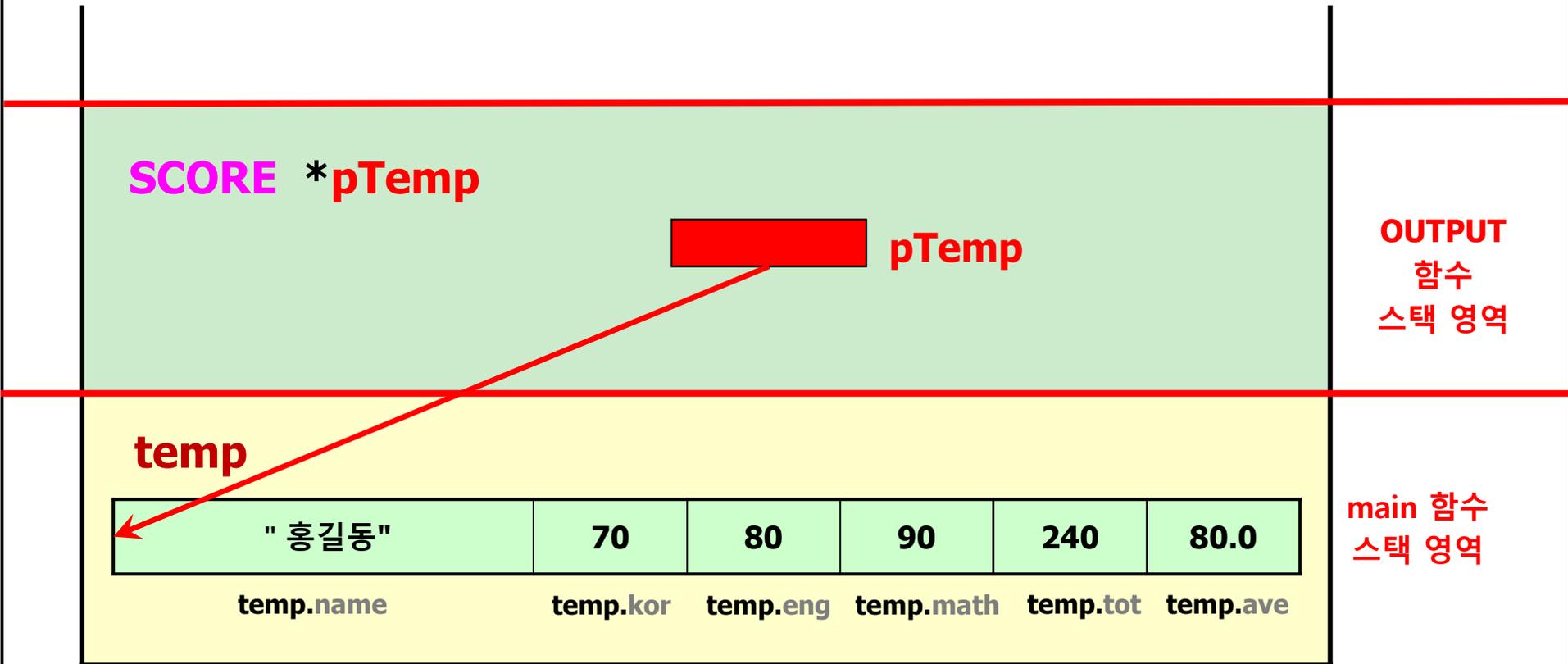
```
        temp.name, temp.kor, temp.eng, temp.math, temp.tot, temp.ave);
```

```
}
```

구조체와 함수 (8/10)

- 구조체의 메모리 주소를 함수에 전달

```
OUTPUT(&temp);
```



구조체와 함수 (9/10)

예제 9-9: 구조체 메모리 주소를 함수에 전달 -- pass by address (1/2)

```
#include <stdio.h>

// 구조체 설계: SCORE(struct _score)
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

// 함수 원형 선언
void input(SCORE *pScore);
void print(SCORE *pScore);

int main(void)
{
    // 구조체 변수 선언
    SCORE s;

    // 학생 정보 입출력
    input(&s);
    print(&s);

    return 0;
}
```



Microsoft Visual Studio 디버그

학생 성적을 입력 하세요...

학생 이름: 홍길동
국어 점수: 70
영어 점수: 80
수학 점수: 91

학생 정보

홍길동 70 80 91 241 80.33

구조체와 함수 (10/10)

예제 9-9: 구조체 메모리 주소를 함수에 전달 -- pass by address (2/2)

// 한 명의 학생 정보 입력

```
void input(SCORE *pS) {
    printf("학생 성적을 입력 하세요. \n\n");
    printf("학생 이름 : ");          gets_s(pS->name, sizeof(pS->name));
    printf("국어 점수 : ");          scanf_s("%d", &pS->kor);
    printf("영어 점수 : ");          scanf_s("%d", &pS->eng);
    printf("수학 점수 : ");          scanf_s("%d", &pS->math);

    pS->tot = pS->kor + pS->eng + pS->math;
    pS->ave = (float)pS->tot / 3;

    return;
}
```

// 한 명의 학생 정보 출력

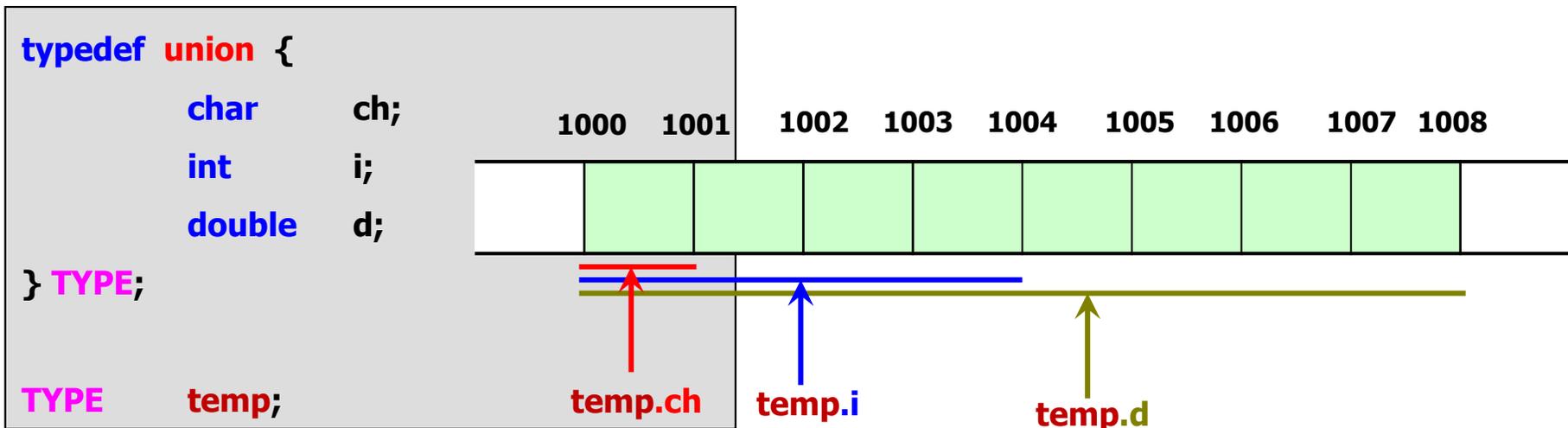
```
void print(SCORE *pS) {
    printf("\n\t ### 입력 정보 ### \n\n");
    printf("%10s %3d %3d %3d %5d %8.2f \n",
           pS->name, pS->kor, pS->eng, pS->math, pS->tot, pS->ave);
}
```


공용체의 이해 (1/4)

● 공용체(Union Type)

○ 메모리에서 서로 다른 데이터 형식을 공유할 수 있는 개념

- 공용체는 구조형과 같은 문법 양식을 따른다.
 - `struct` 키워드 대신 `union` 키워드를 쓰는 것 외에는 모두 똑같다.
 - 공용체를 참조하는 규칙은 구조형을 참조하는 것과 동일하다.



공용체의 이해 (2/4)

예제 9-10: 공용체 -- 형식 정의 및 변수 선언과 데이터 저장

```
#include <stdio.h>

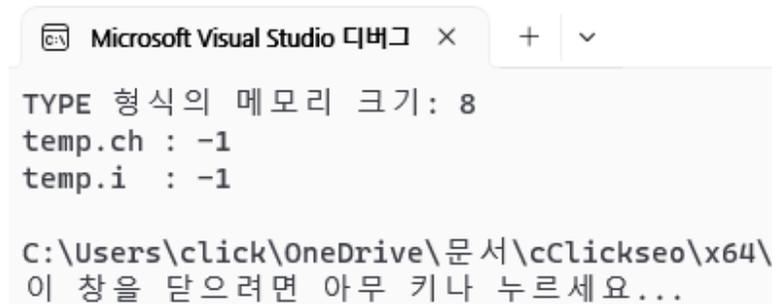
// 공용체 형식 정의
typedef union _type {
    char    ch;
    int     i;
    double  d;
} TYPE;

int main(void)
{
    // 공용체 변수 선언
    TYPE    temp;

    temp.i = 0xFFFFFFFF;

    printf("TYPE 형식의 메모리 크기: %zd \n", sizeof(TYPE) );
    printf("temp.ch : %d \n", temp.ch );
    printf("temp.i   : %d \n", temp.i );

    return 0;
}
```



Microsoft Visual Studio 디버그

TYPE 형식의 메모리 크기: 8
temp.ch : -1
temp.i : -1

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...

공용체의 이해 (3/4)

예제 9-11: 구조체와 공용체 -- "Big endian" 과 "Little endian" (1/2)

```
#include <stdio.h>

// 비트 필드 구조체 형식 정의
typedef struct _uint {
    unsigned int    FirstBit: 8;
    unsigned int    SecondBit: 8;
    unsigned int    ThirdBit: 8;
    unsigned int    ForthBit: 8;
} UINT;

// 공용체 형식 정의: 정수형과 구조체 멤버
typedef union _endian {
    int    a;    // 정수형 멤버
    UINT   b;    // 구조체 멤버
} ENDIAN;
```

```
Microsoft Visual Studio 디버그 x + v
temp.i: 12345678
temp.i: 78
temp.i: 56
temp.i: 34
temp.i: 12

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

공용체의 이해 (4/4)

예제 9-11: 구조체와 공용체 -- "Big endian" 과 "Little endian" (2/2)

```
int main(void)
{
    // 공용체 변수 선언
    ENDIAN      temp;
    temp.a = 0x12345678;

    // "Big endian" 과 "Little endian"
    printf("temp.i: %X \n", temp.a );           // 12345678
    printf("temp.i: %X \n", temp.b.FirstBit ); // 78
    printf("temp.i: %X \n", temp.b.SecondBit ); // 56
    printf("temp.i: %X \n", temp.b.ThirdBit );  // 34
    printf("temp.i: %X \n", temp.b.ForthBit );  // 12

    return 0;
}
```

Microsoft Visual Studio 디버그

```
temp.i: 12345678
temp.i: 78
temp.i: 56
temp.i: 34
temp.i: 12
```



공용체와 열거형

열거형의 이해



열거형의 이해 (1/3)

- 열거형(Enumerated Type)

- 정수형에 기반

- 열거형에서 각 정수 값은 열거형 상수라고 불리는 식별자
- 목적: 정수에 이름을 할당하는 것

```
// 열거화 된 변수형
```

```
enum { 열거형 상수, .... } 변수 식별자;
```

```
// 열거화 된 태그형
```

```
enum TagName { 열거형 상수, ... };
```

```
enum TagName 변수 식별자;
```

열거형의 이해 (2/3)

- 열거형 형식 정의: 태그형, 재정의형

```
// 열거화 된 태그형
```

```
enum Color { RED, WHITE, BLUE, GREEN, YELLOW };  
enum Color aColor;
```

```
// 재정의 된 열거형
```

```
typedef enum { RED, WHITE, BLUE, GREEN, YELLOW } COLOR;  
COLOR aColor;
```

```
// 0부터 시작하여 각 열거 상수에 정수 값 부여
```

```
enum months { jan, feb, mar, apr, may, jun,  
             jul, aug, sep, oct, nov, dec};
```

```
enum months { jan=1, feb, mar, apr, may, jun,  
             jul, aug, sep, oct, nov, dec };
```

```
enum months { jan=1, feb=2, mar=3, apr=4, may=5, jun=6,  
             jul=7, aug=8, sep=9, oct=10, nov=11, dec=12 };
```

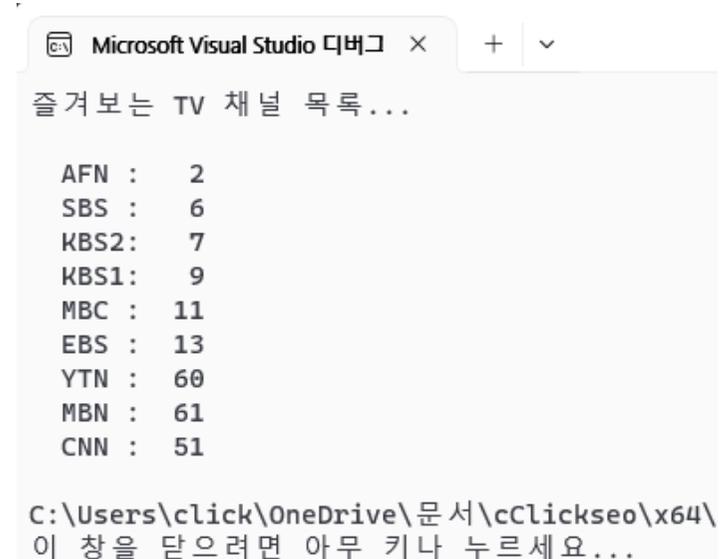
열거형의 이해 (3/3)

예제 9-12: 열거형 -- TV 채널

```
#include <stdio.h>
int main(void)
{
    // 열거형 정의
    enum TV {
        afn = 2, sbs = 6, kbs2 = 7,
        kbs1 = 9, mbc = 11, ebs = 13,
        ytn = 60, mbn = 61, cnn = 51
    };

    printf("즐거보는 TV 채널 목록... \n\n");
    printf("  AFN : %3d\n", afn );
    printf("  SBS : %3d\n", sbs );
    printf("  KBS2: %3d\n", kbs2 );
    printf("  KBS1: %3d\n", kbs1 );
    printf("  MBC : %3d\n", mbc );
    printf("  EBS : %3d\n", ebs );
    printf("  YTN : %3d\n", ytn );
    printf("  MBN : %3d\n", mbn );
    printf("  CNN : %3d\n", cnn );

    return 0;
}
```



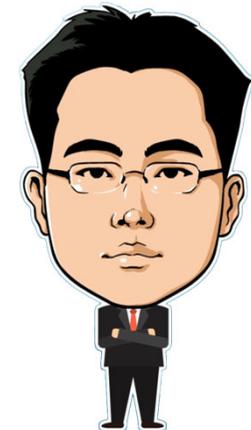
```
Microsoft Visual Studio 디버그 x + v
즐거보는 TV 채널 목록...

AFN : 2
SBS : 6
KBS2: 7
KBS1: 9
MBC : 11
EBS : 13
YTN : 60
MBN : 61
CNN : 51

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

참고문헌

- [1] 서현우, "혼자 공부하는 C 언어: 1:1 과외 하듯 배우는 프로그래밍 자습서", 한빛미디어, 2023.
- [2] Paul Deitel, Harvey Deitel, "C How to Program", Global Edition, 8/E, Pearson, 2016.
- [3] Kamran Amini, 박지윤 번역, "전문가를 위한 C : 동시성, OOP부터 최신 C, 고급 기능까지!", 한빛미디어, 2022.
- [4] 서두옥, "(열혈강의) 또 하나의 C : 프로그래밍은 셀프입니다", 프리렉, 2012.
- [5] Behrouz A. Forouzan, Richard F. Gilberg, 김진 외 7인 공역, "구조적 프로그래밍 기법을 위한 C", 도서출판 인터비전, 2004.
- [6] Brian W. Kernighan, Dennis M. Ritchie, 김석환 외 2인 공역, "The C Programming Language", 2/E, 대영사, 2004.
- [7] "C reference", cppreference.com, 2024 of viewing the site, <https://en.cppreference.com/w/c>.



이 강의자료는 저작권법에 따라 보호받는 저작물이므로 무단 전제와 무단 복제를 금지하며, 내용의 전부 또는 일부를 이용하려면 반드시 저작권자의 서면 동의를 받아야 합니다.

Copyright © Clickseo.com. All rights reserved.

