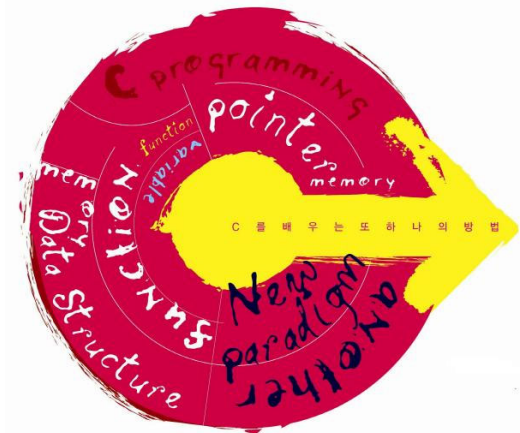


C Programming

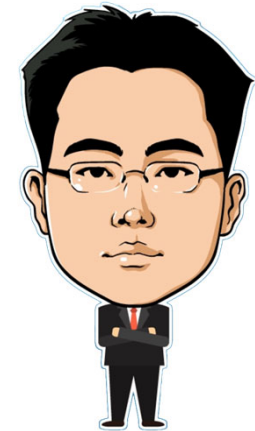
복합 데이터 유형 (Composite Data Types)



Seo, Doo-Ok

Clickseo.com

clickseo@gmail.com



목 차



백문이불여일타(百聞而不如一打)

- 구조체의 이해
- 구조체와 포인터 그리고 함수
- 공용체와 열거형



구조체



백문이불여일타(百聞而不如一打)

- 구조체의 이해

- 구조체 배열

- 중첩 구조체

- 구조체와 포인터 그리고 함수

- 공용체와 열거형



구조체의 이해 (1/8)

- **구조체(Structure)**

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

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

S

구조체의 이해 (2/8)

- 구조체 선언 및 정의

- C 언어에서 구조체 선언 및 정의

1. 구조체형 변수(structure variable)
2. 태그형 구조체(tagged structure)
3. 형식 정의된 구조체(type-defined structure)

```
// 구조체형 변수(structure variable)
```

```
struct {  
    ...  
} variable_identifier;
```

```
// 태그형 구조체(tagged structure)
```

```
struct tagName {  
    ...  
} variable_identifier;  
  
struct tagName variable_identifier;
```

```
// 형식 정의된 구조체(type-defined structure)
```

```
typedef struct {  
    ...  
} NEW_TYPE;  
  
NEW_TYPE variable_identifier;
```

구조체의 이해 (3/8)

- 구조체 선언 및 정의: 구조체형 변수
 - 구조체형 변수

```
// 구조체형 변수(structure variable)
struct {
    ... field-list ...
} variable_identifier;
```

```
struct {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} s;
```

구조체의 이해 (4/8)

- 구조체 선언 및 정의: 태그형 구조체

- 태그형 구조체

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

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

```
struct _score    s;
```

```
void OUTPUT( struct _score temp );
```

```
// 태그형 구조체(tagged structure)
```

```
struct tagName {
```

```
    ...
```

```
} variable_identifier;
```

```
struct tagName variable_identifier;
```

구조체의 이해 (5/8)

- 구조체 선언 및 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

```
typedef struct _score {  
    char    name[12];  
    int     kor, eng, math, tot;  
    float   ave;  
} SCORE;  
  
SCORE    s;  
  
void OUTPUT( SCORE temp );
```

```
// 형식 정의된 구조체(type-defined structure)  
typedef struct {  
    ...  
} NEW_TYPE;  
  
NEW_TYPE    variable_identifier;
```


구조체의 이해 (6/8)

● 구조체 초기화

- 초기값은 중괄호를 사용하며 콤마로 분리
 - 단, 구조체에 정의된 각 구조체 멤버들의 형식과 일치해야 한다.

```
typedef struct _score {  
    char    name[12];  
    int     kor, eng, math, tot;  
    float   ave;  
} SCORE;  
  
SCORE    s = {"서두옥", 70, 80, 90, 240, 80.0};
```

S	"서두옥"	70	80	90	240	80.0
----------	-------	----	----	----	-----	------

구조체의 이해 (7/8)

● 구조체 멤버

○ 구조체 멤버 접근

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

S	"서두옥"	70	80	90	240	80.0
	s.name	s.kor	s.eng	s.math	s.tot	s.ave

○ 구조체의 크기 : 각 멤버들이 차지하는 메모리 용량의 총합

```
sizeof ( struct _score ) = 32 Bytes
```

```
sizeof ( SCORE ) = 32 Bytes
```

구조체의 이해 (8/8)

예제 8-1: 구조체의 이해

```
#include <stdio.h>

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

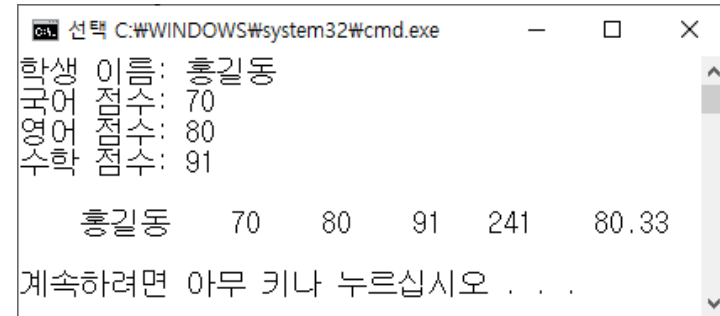
int main(void)
{
    // 구조체형 변수 선언
    SCORE s;

    printf("학생 이름: ");
    printf("국어 점수: ");
    printf("영어 점수: ");
    printf("수학 점수: ");

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

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

    return 0;
}
```



```
선택 C:#WINDOWS#system32#cmd.exe
학생 이름: 홍길동
국어 점수: 70
영어 점수: 80
수학 점수: 91

홍길동   70   80   91   241   80.33

계속하려면 아무 키나 누르십시오 . . .
```



구조체의 이해

구조체 배열, 중첩 구조체



구조체 배열 (1/3)

- 구조체 배열

```
typedef struct _score {  
    char    name[12];  
    int     kor, eng, math, tot;  
    float   ave;  
} SCORE;  
  
SCORE arr[3];
```

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

arr

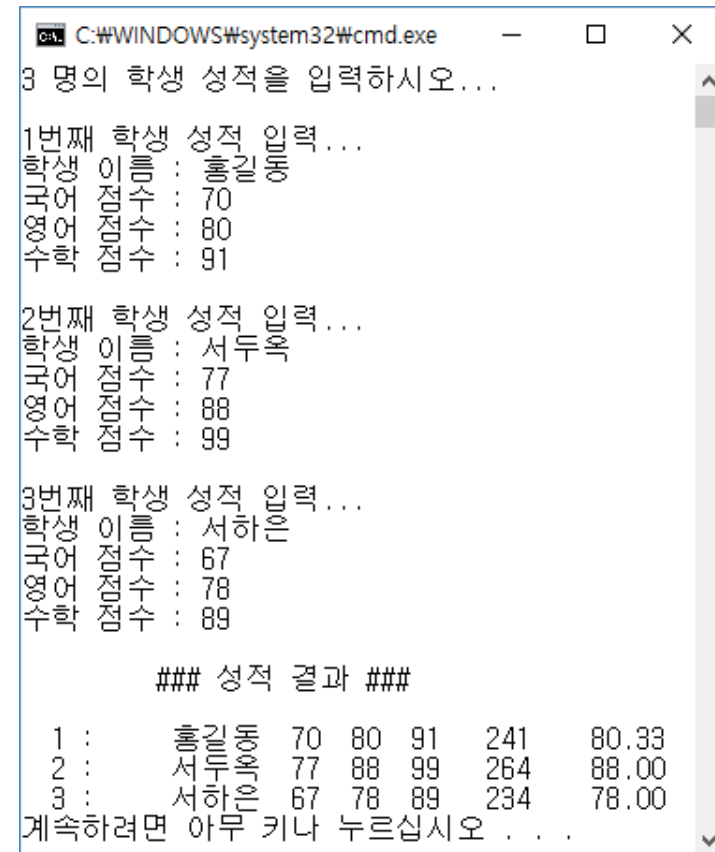
구조체 배열 (2/3)

예제 8-2: 구조체 배열

(1/2)

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

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



```
C:\WINDOWS\system32\cmd.exe
3 명의 학생 성적을 입력하시오...

1번째 학생 성적 입력...
학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 91

2번째 학생 성적 입력...
학생 이름 : 서두옥
국어 점수 : 77
영어 점수 : 88
수학 점수 : 99

3번째 학생 성적 입력...
학생 이름 : 서하은
국어 점수 : 67
영어 점수 : 78
수학 점수 : 89

### 성적 결과 ###

1 :   홍길동 70 80 91 241 80.33
2 :   서두옥 77 88 99 264 88.00
3 :   서하은 67 78 89 234 78.00
계속하려면 아무 키나 누르십시오 . . .
```

구조체 배열 (3/3)

예제 8-2: 구조체 배열

(2/2)

```
int main(void)
{
    SCORE arr[stuMAXSIZE];

    printf("%d 명의 학생 성적을 입력하시오... \n", stuMAXSIZE );
    for(int i = 0; i<stuMAXSIZE; i++) {
        printf("\n%d번째 학생 성적 입력... \n", i + 1);
        printf("학생 이름 : "); gets_s( arr[i].name , sizeof(arr[i].name) );
        printf("국어 점수 : "); scanf_s("%d", &arr[i].kor );
        printf("영어 점수 : "); scanf_s("%d", &arr[i].eng );
        printf("수학 점수 : "); scanf_s("%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;
    }

    printf("\n\t ### 성적 결과 ### \n\n");
    for(int i = 0; i<stuMAXSIZE; 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;
}
```

중첩 구조체

- 중첩 구조체

```
typedef struct {  
    struct {  
        int year;  
        int month;  
        int day;  
    } date;  
    struct {  
        int hour;  
        int min;  
        int sec;  
    } time;  
} STAMP;  
  
STAMP stamp;
```

추천하지 않는 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;  
} STAMP;  
  
STAMP stamp;
```

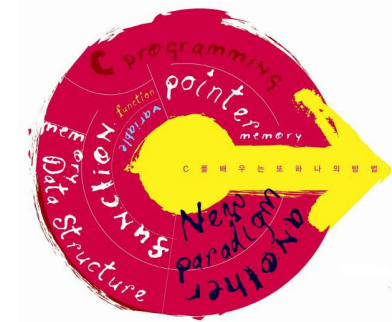
선호하는 방식

구조체와 포인터 그리고 함수



백문이불여일타(百聞而不如一打)

- 구조체의 이해
- 구조체와 포인터 그리고 함수
 - 구조체와 포인터
 - 구조체와 함수
- 공용체와 열거형



구조체와 포인터 (1/2)

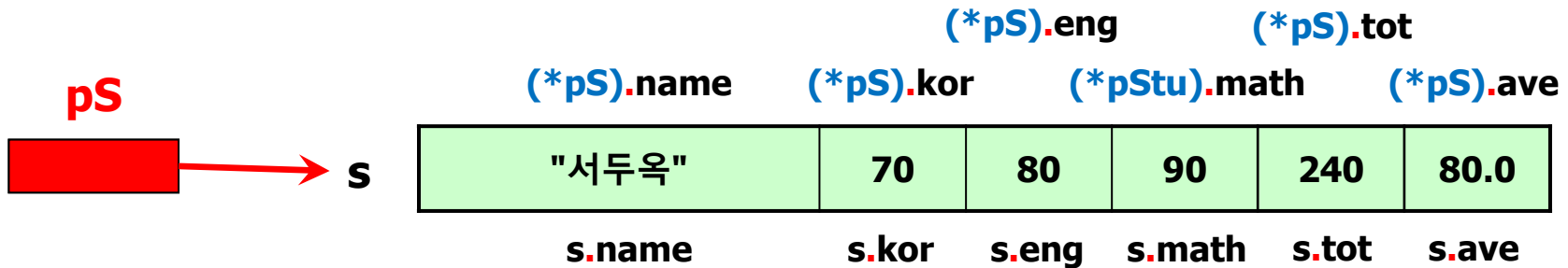
- 간접 표기법(Indirection notation)

s.name

(*pS).name

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

SCORE  s;
SCORE  *pS = &s;
```



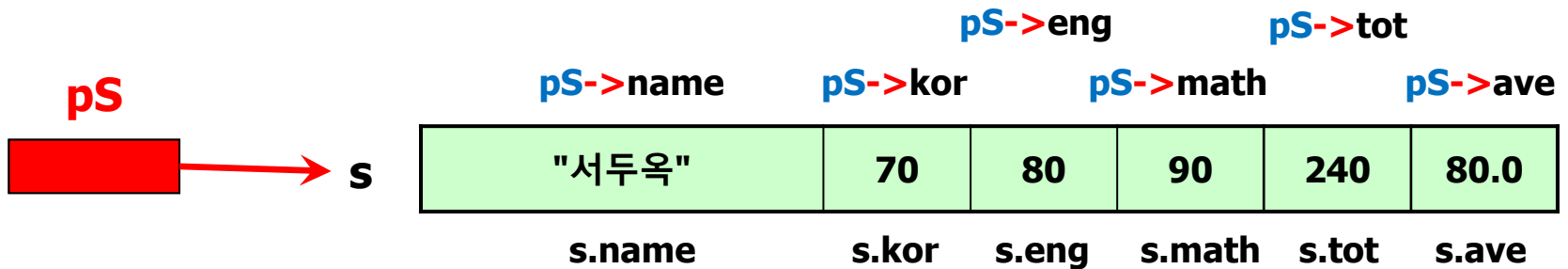
구조체와 포인터 (2/2)

- 선택 표기법(Selection notation)

s.name

pS->name

```
typedef struct _score {  
    char    name[12];  
    int     kor, eng, math, tot;  
    float   ave;  
} SCORE;  
  
SCORE    s;  
SCORE    *pS = &s;
```





구조체와 포인터 그리고 함수

구조체와 함수



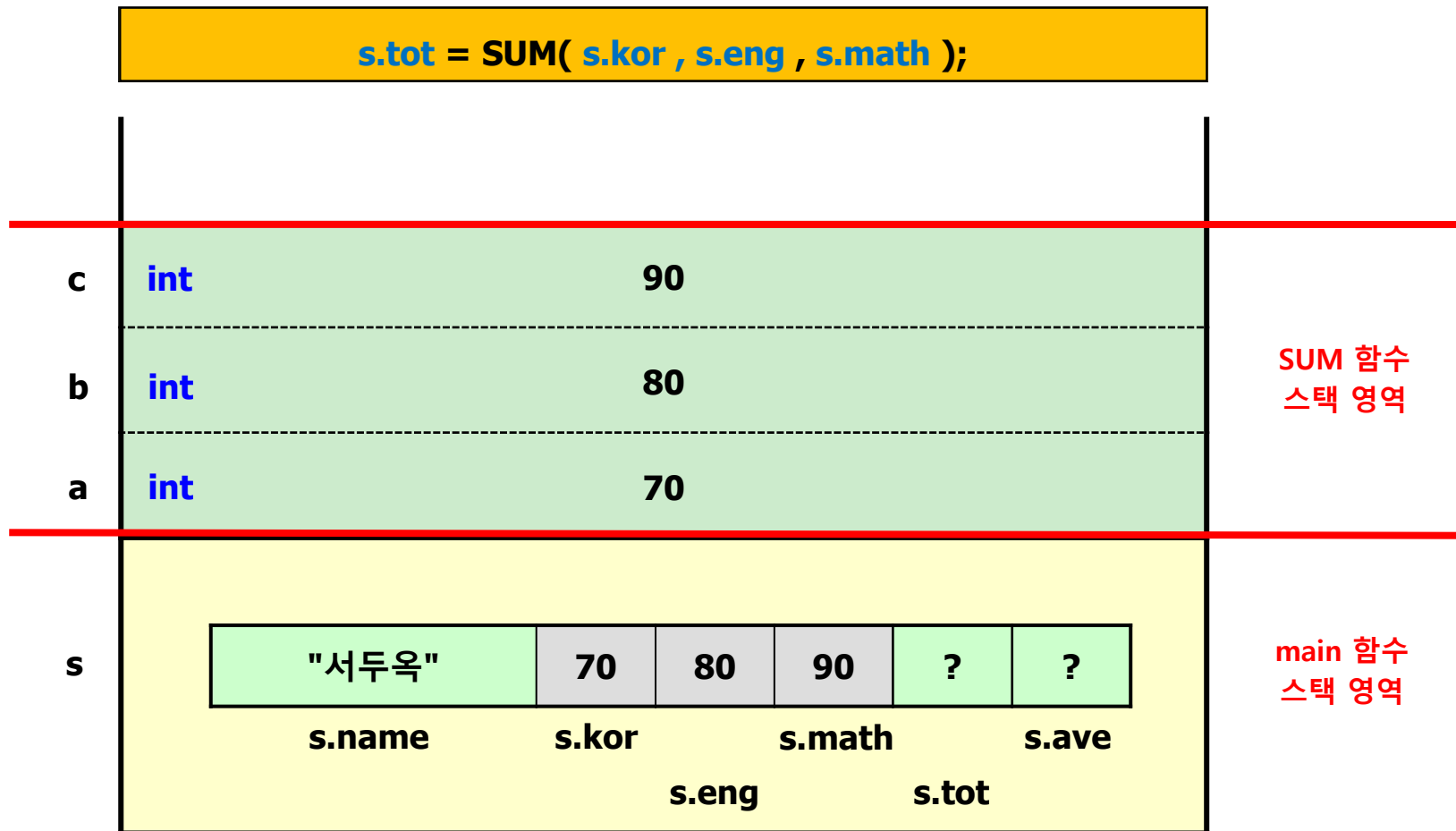
구조체와 함수 (1/10)

- **함수에서 구조체 멤버 접근 방법**

1. **구조체 멤버를 함수에 전달**하고(pass by value), 함수 내부에서 일반 변수처럼 접근한다.
2. **구조체 전체를 함수에 전달**하고(pass by value), 함수 내부에서 각 멤버들을 일반 변수처럼 접근한다.
3. **구조체 메모리 주소를 함수에 전달**하고(pass by address), 함수 내부에서 **간접 접근 방법**(간접 표기법 또는 선택 표기법)을 이용하여 구조체 멤버를 접근한다.

구조체와 함수 (2/10)

- 구조체 멤버를 함수에 전달



구조체와 함수 (3/10)

예제 8-3: 구조체 멤버를 함수에 전달 -- pass by value

(1/2)

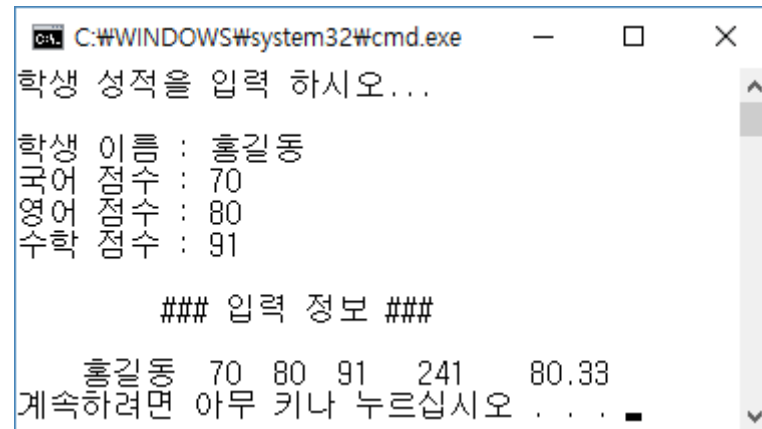
```
#include <stdio.h>

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

int     SUM( int, int, int );
float   AVE( int, int, int );

int main(void)
{
    SCORE s;

    printf("학생 성적을 입력 하시오... \n\n");
    printf("학생 이름 : ");
    printf("국어 점수 : ");
    printf("영어 점수 : ");
    printf("수학 점수 : ");
```



```
C:\WINDOWS\system32\cmd.exe
학생 성적을 입력 하시오...
학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 91

### 입력 정보 ###
홍길동 70 80 91 241 80.33
계속하려면 아무 키나 누르십시오...
```

구조체와 함수 (4/10)

예제 8-3: 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 ### 입력 정보 ### \n\n");
printf("%10s %3d %3d %3d %5d %8.2f \n",
       s.name, s.kor, s.eng, s.math, s.tot, s.ave );

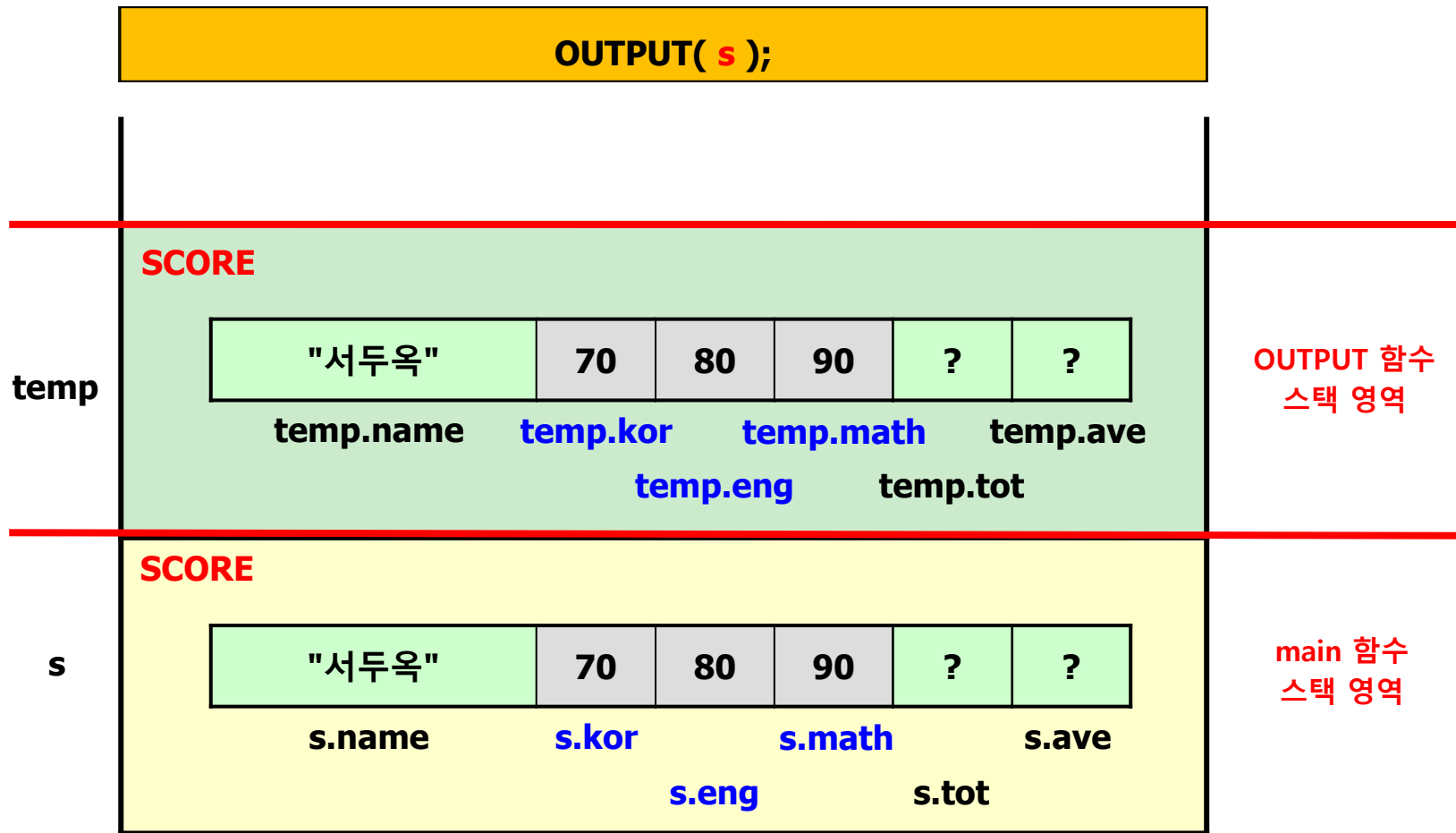
return 0;
}

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;
}
```


구조체와 함수 (5/10)

- 구조체 전체를 함수에 전달



구조체와 함수 (6/10)

예제 8-4: 구조체 전체를 함수에 전달 -- pass by value

(1/2)

```
#include <stdio.h>

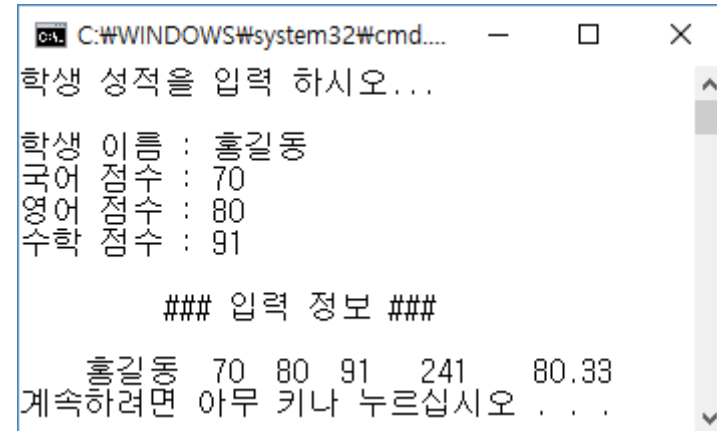
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

SCORE INPUT(void);
void OUTPUT(SCORE);

int main(void)
{
    SCORE s;

    s = INPUT();
    OUTPUT(s);

    return 0;
}
```



```
C:\WINDOWS\system32\cmd...
학생 성적을 입력 하시오...
학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 91

### 입력 정보 ###

홍길동 70 80 91 241 80.33
계속하려면 아무 키나 누르십시오 ...
```

구조체와 함수 (7/10)

예제 8-4: 구조체 전체를 함수에 전달 -- pass by value

(2/2)

SCORE INPUT(**void**)

{

SCORE temp;

printf("학생 성적을 입력 하시오... \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;

}

void OUTPUT(**SCORE** temp)

{

printf("\n\t ### 입력 정보 ### \n\n");

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

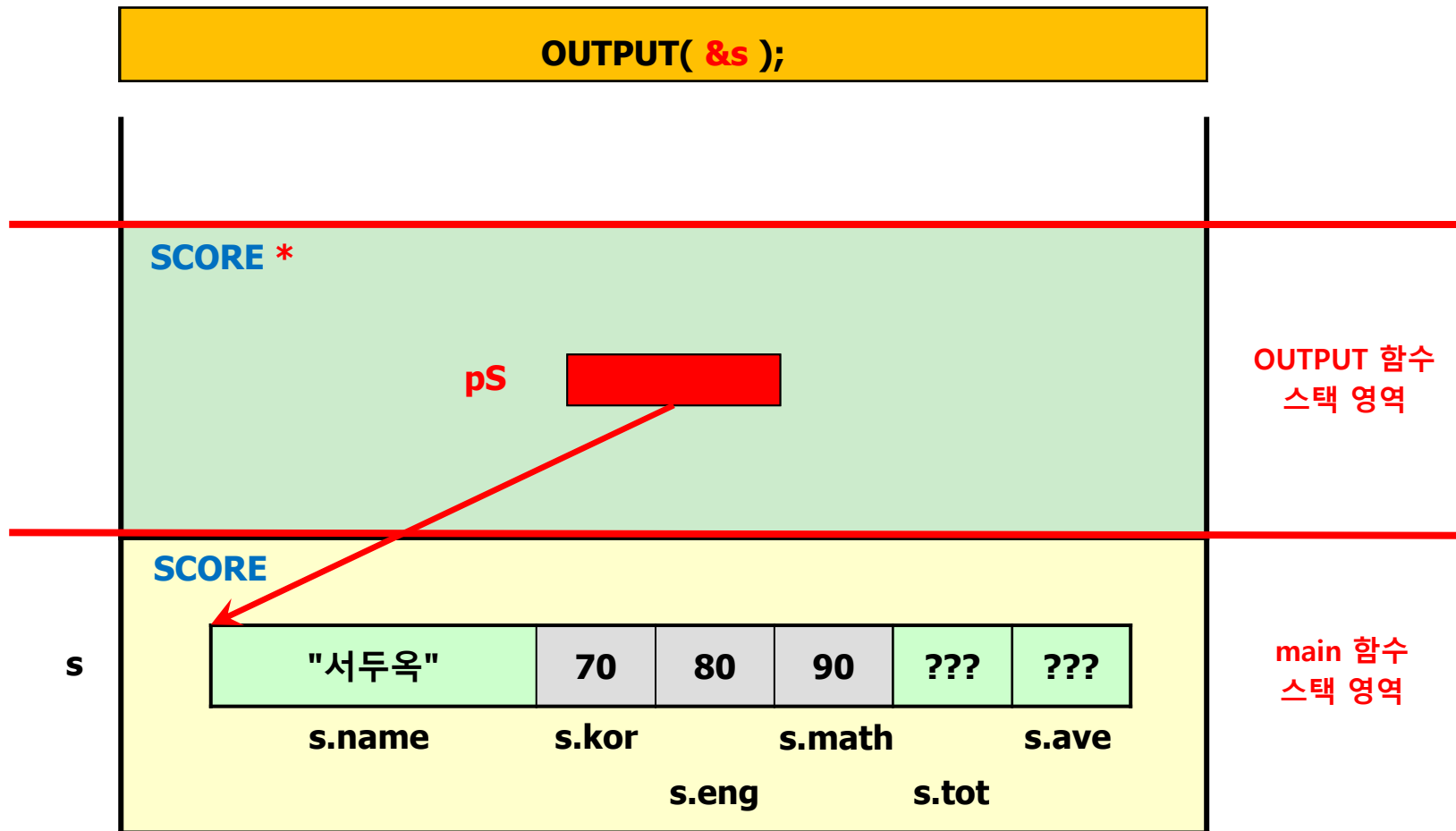
temp.name, temp.kor, temp.eng, temp.math, temp.tot, temp.ave);

return;

}

구조체와 함수 (8/10)

- 구조체 메모리 주소를 함수에 전달



구조체와 함수 (9/10)

예제 8-5: 구조체 메모리 주소를 함수에 전달 -- pass by address (1/2)

```
#include <stdio.h>

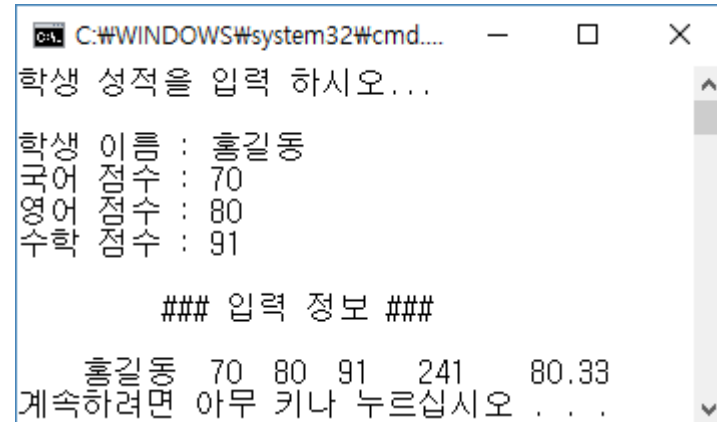
typedef struct _score {
    char    name[12];
    int     kor, eng, math, tot;
    float   ave;
} SCORE;

void INPUT( SCORE * );
void OUTPUT( SCORE * );

int main(void)
{
    SCORE s;

    INPUT( &s );
    OUTPUT( &s );

    return 0;
}
```



```
C:\WINDOWS\system32\cmd...
학생 성적을 입력 하시오...
학생 이름 : 홍길동
국어 점수 : 70
영어 점수 : 80
수학 점수 : 91

### 입력 정보 ###
홍길동 70 80 91 241 80.33
계속하려면 아무 키나 누르십시오 ...
```

구조체와 함수 (10/10)

예제 8-5: 구조체 메모리 주소를 함수에 전달 -- pass by address (2/2)

```
void INPUT( SCORE *pS )
{
    printf("학생 성적을 입력 하시오... \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;
}

void OUTPUT( 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 );

    return;
}
```

공용체와 열거형



- 구조체의 이해

백문이불여일타(百聞而不如一打)

- 구조체와 포인터 그리고 함수

- **공용체와 열거형**

- 공용체

- 열거형

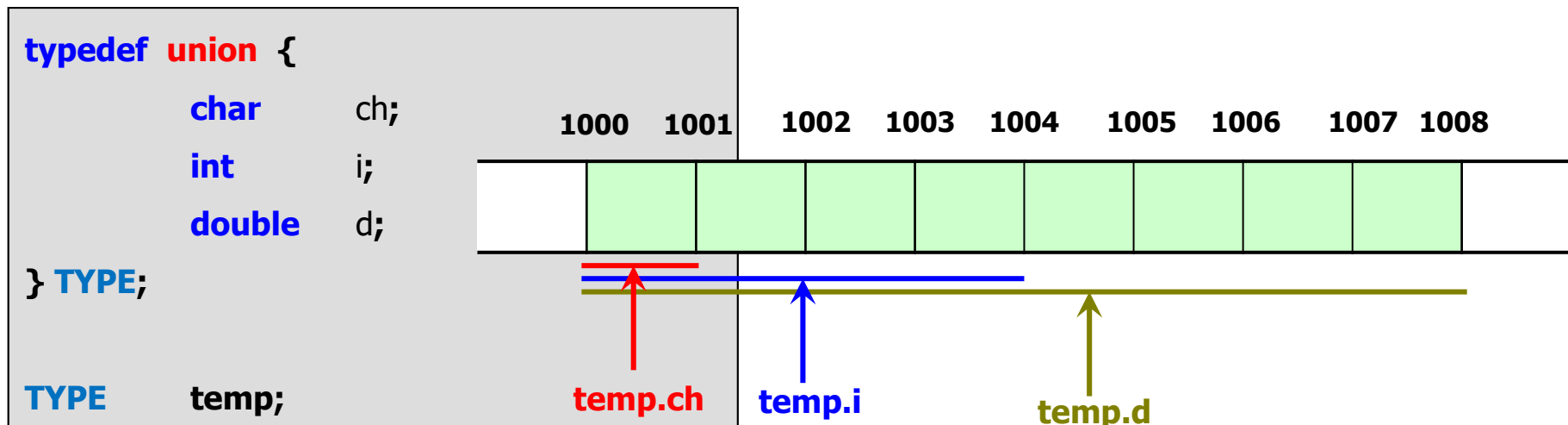


공용체 (1/4)

● 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.



공용체 (2/4)

예제 8-6: 공용체 -- 메모리 할당과 데이터 저장

```
#include <stdio.h>

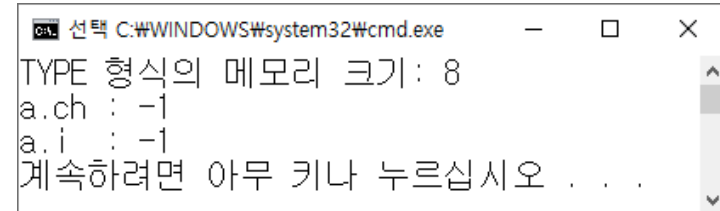
typedef union _type {
    char    ch;
    int     i;
    double  d;
} TYPE;

int main(void)
{
    TYPE    a;

    a.i = 0xFFFFFFFF;

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

    return 0;
}
```



```
선택 C:#WINDOWS#system32#cmd.exe
TYPE 형식의 메모리 크기: 8
a.ch : -1
a.i : -1
계속하려면 아무 키나 누르십시오 . . .
```

공용체 (3/4)

예제 8-7: 구조체와 공용체

(1/2)

```
#include <stdio.h>

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

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

공용체 (4/4)

예제 8-7: 구조체와 공용체

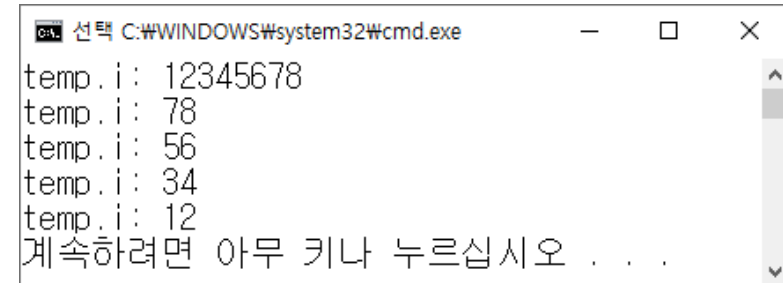
(2/2)

```
int main(void)
{
    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;
}
```



```
선택 C:#WINDOWS#system32#cmd.exe
temp.i: 12345678
temp.i: 78
temp.i: 56
temp.i: 34
temp.i: 12
계속하려면 아무 키나 누르십시오 . . .
```



공용체와 열거형

열거형



열거형 (1/5)

- 열거형(Enumerated Type)

- 정수형에 기반

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

```
// 열거화 된 변수형
enum { 열거형 상수, ... } 변수 식별자;

// 열거화 된 태그형
enum tag { 열거형 상수, ... };
enum tag 변수 식별자;
```

열거형 (3/5)

- 열거형의 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;
```

열거형 (4/5)

- 열거형의 정의: 초기화

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

- 0 부터 시작하여 각 열거 상수에 정수 값 부여

```
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 };
```

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

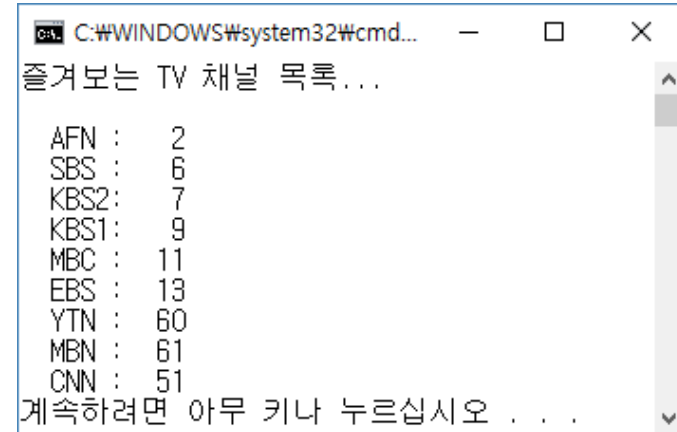
공용체 (5/5)

예제 8-8: 열거형 -- TV 채널

```
#include <stdio.h>
int main(void)
{
    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;
}
```

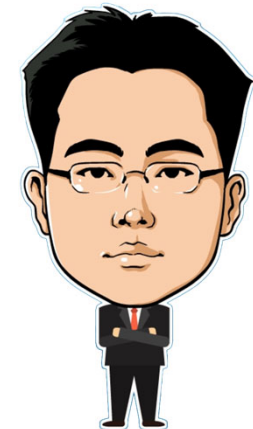


```
C:\WINDOWS\system32\cmd...
즐거보는 TV 채널 목록...

AFN : 2
SBS : 6
KBS2: 7
KBS1: 9
MBC : 11
EBS : 13
YTN : 60
MBN : 61
CNN : 51
계속하려면 아무 키나 누르십시오 . . .
```


참고문헌

- [1] 서두옥, 이동호(감수), (열혈강의)"또 하나의 C : 프로그래밍은 셀프입니다", 프리렉, 2012.
- [2] Paul Deitel, Harvey Deitel, "C How to Program", Global Edition, 8/E, Pearson, 2016.
- [3] SAMUEL P. HARBISON III, GUY L. STEELE, "C 프로그래밍 언어, C : A Reference Manual", 5/E, Pearson Education Korea, 2005.
- [4] 문병로, "쉽게 배우는 알고리즘 - 관계 중심의 사고법", 개정판, 한빛아카데미, 2018.
- [5] 주우석, "C·C++ 로 배우는 자료구조론", 한빛아카데미, 2015.
- [6] Behrouz A. Forouzan, Richard F. Gilberg, 김진 외 7인 공역, "구조적 프로그래밍 기법을 위한 C", 도서출판 인터비전, 2004.
- [7] Brian W. Kernighan, Dennis M. Ritchie, 김석환 외 2인 공역, "The C Programming Language", 2/E, 대영사, 2004.
- [8] 김일광, "C 프로그래밍 입문 : 프로그래밍을 모국어처럼 유창하게", 한빛미디어, 2004.
- [9] 정재은, "다시 체계적으로 배우는 C 언어 포인터", 정보문화사, 2003.



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.