

# C 프로그래밍 #1

# hello.c

```
/* 20104001 홍길동 03050900 */
```

파일 생성 날짜  
및 시간

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

```
int main(void)
```

```
{
```

```
    printf("Hello, world !\n");
```

```
    return 0;
```

```
}
```

TAB 키를 이용하여  
정렬

# hello.c 에서 확인 사항(1)

## ■ Comment

- `/* */`
- `//`
- 프로그램에 대한 개요 설명을 반드시 포함

## ■ Preprocessor

- `#`
- `#include`

## ■ 함수의 형태

```
return_type function_name ( parameter_list )  
{  
  
    return ;  
}
```

# hello.c 에서 확인 사항(2)

- type : int, short, long, signed, unsigned, float, double, char
- ; 의 역할 : 문장 구분
- 함수 호출의 의미
- printf()의 기능
  - 변환 문자 (c, d, f, s, ...)
- return의 역할

# sec\_to\_hms.c

```
/* 20034201 홍길동 03050900 */
#include <stdio.h>

#define SECS_PER_HOUR      3600
#define SECS_PER_MIN      60

int main(void)
{
    int    total_secs;
    int    hours, mins, secs;
    int    remain;

    printf("total secs ? ");
    scanf("%d", &total_secs);

    hours = total_secs / SECS_PER_HOUR;
    remain = total_secs % SECS_PER_HOUR;
    mins = remain / SECS_PER_MIN;
    secs = remain % SECS_PER_MIN;

    printf("%d secs -> %d hour(s), %d min(s), %d sec(s)\n", total_secs, hours, mins, secs);

    return 0;
}
```

# sec\_to\_hms.c에서 확인 사항(1)

- #define을 이용하는 이유
  - 프로그램의 이해를 돕기 이해
  - 프로그램 수정이 용이
- 변수와 상수 개념
- 식별자(identifier)
- 배정문(할당문)의 의미
  - lvalue, rvalue
  - 다른 언어 예 :=
  - ==

# sec\_to\_hms.c에서 확인 사항(2)

- 정수와 실수
- 정수형 연산자
  - +, -, \*, /, %
- scanf()의 기능
  - &변수이름의 의미
  - scanf()의 결과 값의 의미
  - 변환 문자 (c, d, f, ld, ...)
- scanf() 이용 패턴
  - 사용자에게 입력 받을 내용을 먼저 안내

# 분기문 - if(1)

- if 사용 예

```
if (number == 7) {  
    printf("LUCKY");  
}
```

- if-else 사용 예

```
if (score > 59) {  
    printf("PASS");  
}  
else {  
    printf("FAIL");  
}
```



## 분기문 - if(2)

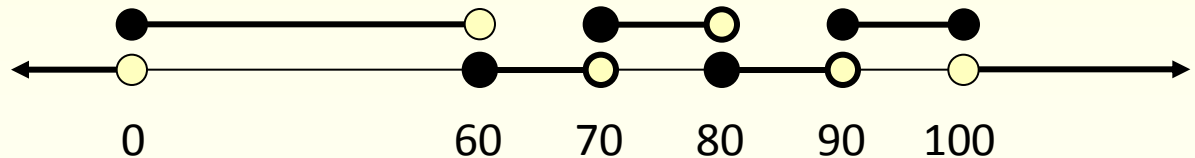
- nested if

```
if (month == 2) {  
    if (leap_year) {  
        days = 29;  
    }  
    else {  
        days = 28;  
    }  
}
```

# 분기문 - if(3)

## ■ if-else if-else 사용 예

```
if (score > 100 || score < 0) {  
    printf("Error");  
}  
else if (score > 89){  
    printf("A");  
}  
else if (score > 79){  
    printf("B");  
}  
else if (score > 69){  
    printf("C");  
}  
else if (score > 59){  
    printf("D");  
}  
Else {  
    printf("F");  
}
```



# 분기문 - switch

- 사용 예

```
switch (choice) {  
    case 'a' :  
    case 'A' :  
        ++a_cnt;  
        break;  
    case 'b' :  
    case 'B' :  
        ++b_cnt;  
        break;  
    default :  
        ++other_cnt;  
}
```

- switch( 정수형 변수 )
- break 문의 역할
- default 의 역할

# 분기문 - ? :

- `expr ? expr : expr`

```
x = (y < z) ? y : z;
```

- 같은 내용

```
if (y < z)
```

```
    x = y;
```

```
else
```

```
    x = z;
```

- 사용 예

```
printf(“%d min%c\n”, mins, (mins > 1) ? ‘s’ : ‘ ’);
```

# 반복문 - while

- 이용 예

```
while (--n > 0) {  
    ...  
}
```

- 초기화 문장, 조건문, 조건 값을 변화하는 문장으로 구성

```
n = 0;  
while (n < 10) {  
    ...  
    n++;  
}
```

- 무한 루프와 널(null) 문장

```
while(1) ;
```

# 반복문 - for

- 이용 예

```
for (n = 0; n < 10; n++) {  
    sum += n;  
}
```

- 초기화 문장, 조건문, 본문 수행 후 수행할 문장으로 구성

- 다른 예(1)

```
n = 0;  
for (; n < 10;) {  
    sum += n;  
    n++;  
}
```

- 다른 예(2)

```
for (n = 0; n < 10; sum += n, n++) ;
```

# 반복문 - do

- 이용 예

```
do {
```

```
...
```

```
} while (--n > 0);
```

- while문과의 차이점

- 0+ 와 1+

# break문과 continue문

- break문
  - switch문
  - for, while, do 반복문 안에서
- continue문
  - for, while, do 반복문 안에서
- 두 문장 모두 적절하게 사용하여야 하며, switch 문에서 쓰이는 break문을 제외한 나머지 경우는 사용 않도록 권고



# err\_sec\_to\_hms.c

```
/* 20034201 홍길동 03050900 */
```

```
#include <stdio.h> ; (1)
```

```
#define SECS_PER_HOUR 3600
```

```
#define SECS_PER_MIN 60; (2)
```

```
int main(void); (3)
```

```
{
```

```
    int    total_secs;
```

```
    int    hours, mins, secs;
```

```
    int    remain;
```

```
{ (4)
```

```
    printf("total secs ?_); (5)
```

```
    scanf("%d_", &_total_secs); (6) (7)
```

```
    hours = total_secs / 3600.; (8)
```

```
    remain = total_secs % SECS_PER_HOUR;
```

```
    mins = remain / SECS_PER_MIN;
```

```
    secs = remain % SECS_PER_MIN;
```

```
    printf("%d secs -> %d hour(s), %d min(s), %d sec(s)\n", total_secs, hours, ____ secs); (9)
```

```
    return 0;
```

```
}
```

# 과제

- `err_sec_to_hms.c` 컴파일 및 실행 과정에서 나타나는 오류메시지, 증상 정리
  - 오류 없는 `sec_to_hms.c` 파일에서 오류를 하나씩 생성하면서 컴파일 과정 또는 실행 과정에서 나타나는 메시지와 증상 정리
  - 실제 컴파일 과정이나 실행 과정에서 오류가 나타나지 않는 경우도 있음
  - 앞 예 외에도 다양한 오류를 만들어 실험
  - 오류 메시지와 오류의 관계를 파일로 정리
- $ax^2 + bx + c = 0$  방정식의  $a, b, c$  값을 입력 받아 근의 성질을 출력하는 프로그램을 작성하시오.
  - 이때 다음 경우를 모두 고려하여야 함.
    - $a = 0, b = 0, c = 0$
    - $a = 0, b = 0, c \neq 0$
    - $a = 0, b \neq 0$
    - 판별식  $> 0$
    - 판별식  $= 0$
    - 판별식  $< 0$
  - 컴파일 후 실행파일을 직접 실행하여 모든 경우에 대해 정상적으로 동작하는지 확인
  - 6가지 경우를 확인한 화면을 캡처해서 과제 제출