

## -Part1-

# 제2장 C 언어의 기본 구조와 표준 입출력

(교재 25페이지 ~ 56페이지)

## 학습목차

**2.1 C언어의 기본 구조**

-교재 27페이지 -

**2.2 Hello C world**

-교재 34페이지 -

**2.3 모니터에 데이터 출력 - printf() 함수**

-교재 35페이지 -

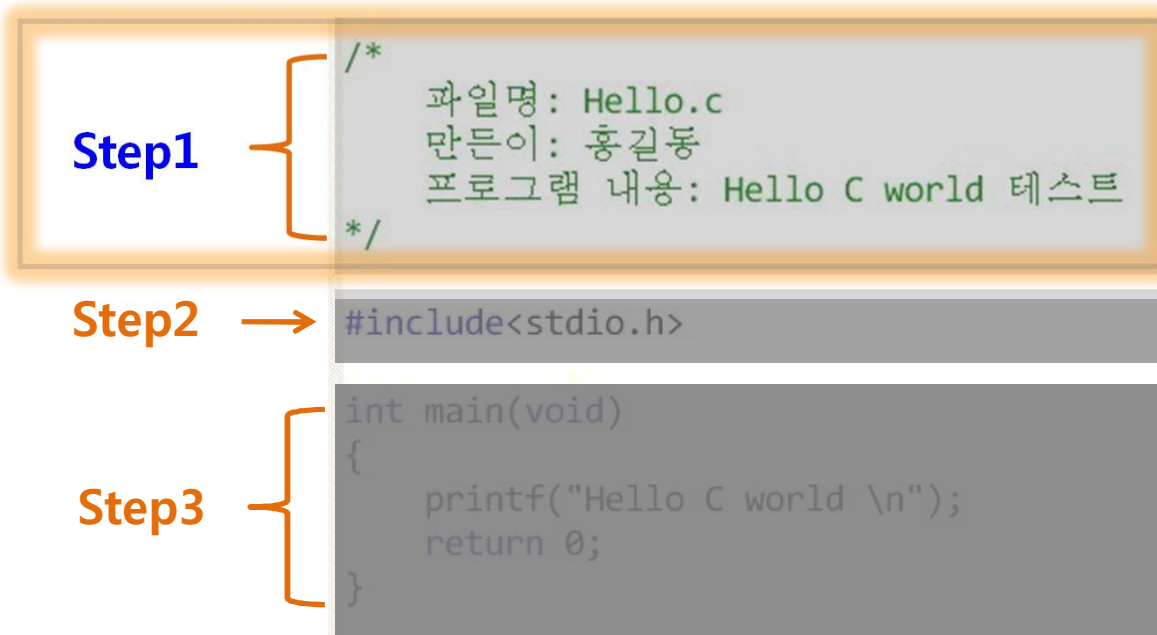
**2.4 키보드로 데이터 입력 scanf()함수**

-교재 47페이지 -

## 2.1 C언어의 기본 구조

-교재 27페이지 -

## 2.1 C언어의 기본 구조 - Step1~Step3 (1/15)



## 2.1 C언어의 기본 구조 - Step1~Step3 (2/15)

### ▶ Step1 - 주석

- ① 주석(Comment): 메모의 기능
- ② 컴파일러는 주석의 내용을 무시

### ③ 주석 처리 방법

```
/*  
파일명: Hello.c  
만든이: 홍길동  
프로그램 내용: Hello C world 테스트  
*/
```

↑  
여러 줄 주석 처리

```
// 파일명: Hello.c  
// 만든이: 홍길동  
// 프로그램 내용: Hello C world 테스트
```

↑  
한 줄 주석 처리

## 2.1 C언어의 기본 구조 - Step1~Step3 (3/15)

### ▶ Step1 - 주석

#### ④ 주석 처리 시 주의 사항 - 중복 사용 금지

```
/*  
  /* 파일명: Hello.c  
    만든이: 홍길동  
    프로그램 내용: Hello C world 테스트 */  
*/
```



오류 발생

#### ⑤ 주석 중복 오류 해결 방법 - // 사용

```
/*  
  // 파일명: Hello.c  
  // 만든이: 홍길동  
  // 프로그램 내용: Hello C world 테스트  
*/
```



정상

## 2.1 C언어의 기본 구조 - Step1~Step3 (4/15)

### ▶ Step1 - 주석

```
/*
파일명 : Hello.c
/* 만든이 : 홍길동 */
프로그램 내용 : Hello C world 테스트
*/
```

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

```
int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```



```
/* /* */ */
주석 중복
오류 발생
```

```
/*
파일명 : Hello.c
// 만든이 : 홍길동
프로그램 내용 : Hello C world 테스트
*/
```

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

```
int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```



```
/* // */
주석 중복
오류 해결
```

## 2.1 C언어의 기본 구조 - Step1~Step3 (5/15)

교재 30 페이지

Step 1

```
/*  
    파일명: Hello.c  
    만든이: 홍길동  
    프로그램 내용: Hello C world 테스트  
*/
```

Step 2

→ #include<stdio.h>

Step 3

```
int main(void)  
{  
    printf("Hello C world \n");  
    return 0;  
}
```



## 2.1 C언어의 기본 구조 - Step1~Step3 (6/15)

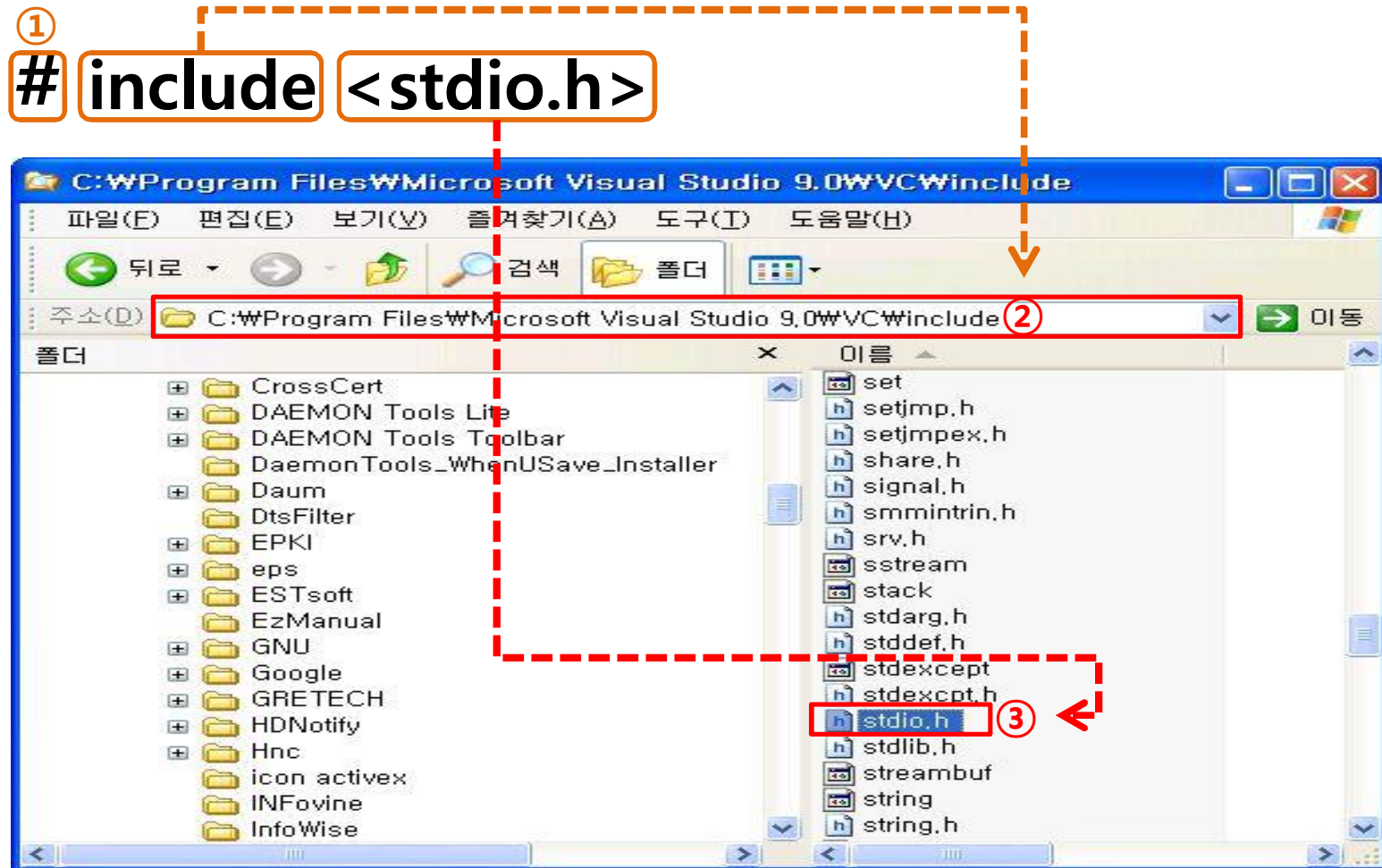
### ▶ Step2 - 전처리기와 헤더파일

① ② ③  
`#include <stdio.h>`

- ① 전처리기(preprocessor)- # : 컴파일을 수행하기 전에 먼저 처리하는 기호
- ② **include** : '포함하다' 라는 뜻을 가지며, 전처리를 지시
- ③ 헤더파일(Header File) : 확장자 .h를 가지는 파일
  - **stdio** 의미 : **S**tandard **I**nput **O**utput (표준 입력 출력)
  - **stdio.h** 의미 : 표준 입력 출력 함수들을 가지고 있는 헤더 파일

## 2.1 C언어의 기본 구조 - Step1~Step3 (7/15)

### ▶ Step2 - 전처리기와 헤더파일



## 2.1 C언어의 기본 구조 - Step1~Step3 (8/15)

### ▶ Step2 - 전처리기와 헤더파일

① # ② include ③ <stdio.h>

- ① 주어: 전처리기 #은
- ② 동사: 포함한다.
- ③ 목적어: stdio.h를

→ 해석: 전처리기야 stdio.h를 포함하는 작업을 먼저 해다오!!

## 2.1 C언어의 기본 구조 - Step1~Step3 (9/15)

### 교재 31 페이지

Step 1

```
/*  
    파일명: Hello.c  
    만든이: 홍길동  
    프로그램 내용: Hello C world 테스트  
*/
```

Step 2 →

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

Step 3

```
int main(void)  
{  
    printf("Hello C world \n");  
    return 0;  
}
```

## 2.1 C언어의 기본 구조 - Step1~Step3 (10/15)

### ▶ Step3 - main() 함수

① main() 함수의 호출과 종료는 누가 결정 하는가?

- 운영체제에 의해서 맨 처음 호출 되고 맨 나중에 종료



```
int main (void)
```

```
{
```

```
    printf ("Hello C world \n");
```

```
    return 0;
```

```
}
```

## 2.1 C언어의 기본 구조 - Step1~Step3 (11/15)

### ▶ Step3 – main() 함수

```
int main (void)
```

```
{
```

```
    printf ("Hello C world \n");
```

```
    return 0;
```

```
}
```



출력 형태    함수 이름    입력 형태



```
int main (void)
```

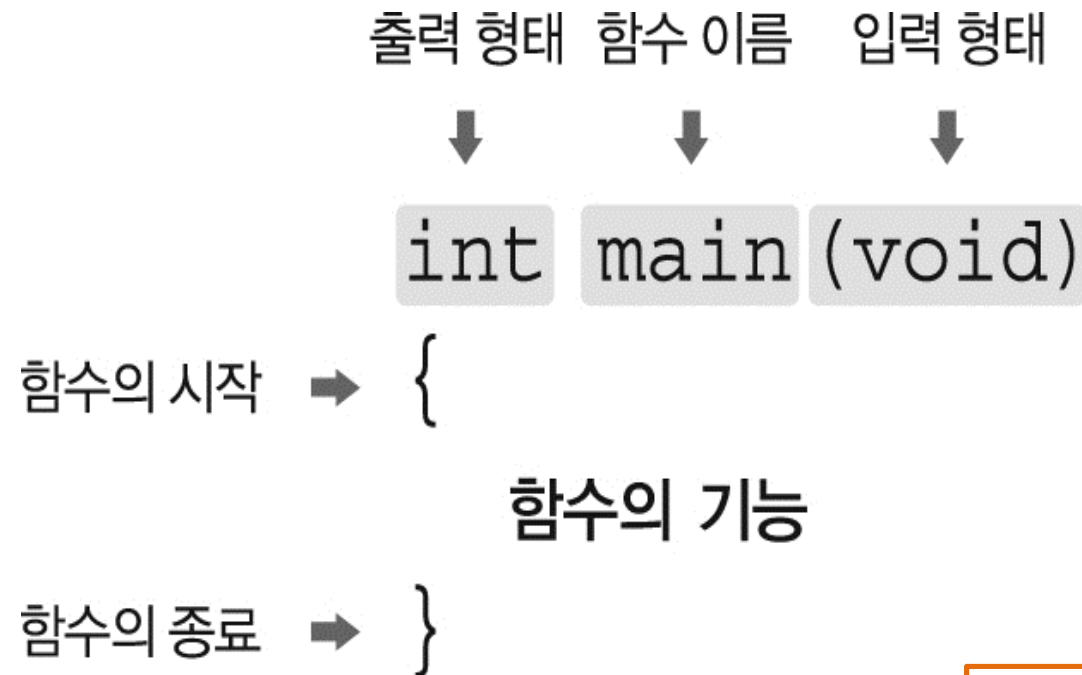
함수의 시작 → {

함수의 기능

함수의 종료 → }

## 2.1 C언어의 기본 구조 - Step1~Step3 (12/15)

### ▶ Step3 – main() 함수



**int** : integer(정수)  
**void** : 빈 공간

## 2.1 C언어의 기본 구조 - Step1~Step3 (13/15)

### ▶ Step3 – main() 함수

② 표준 라이브러리 함수와 헤더파일은 프로그래밍을 편하게 한다.

```
#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```



## 2.1 C언어의 기본 구조 - Step1~Step3 (14/15)

### ▶ Step3 - main() 함수

③ 세미콜론( ; ) 은 문장의 끝을 의미하는 마침표와 같은 존재이다.

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

```
int main(void)
```

```
{
```

```
    printf("Hello C world \n");
```

```
    return 0;
```

```
}
```

세미콜론

## 2.1 C언어의 기본 구조 - Step1~Step3 (15/15)

### ▶ Step3 – main() 함수

④ return은 반환과 종료의 의미를 갖는다.

```
#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```

운영체제에게 0을 반환 하고,  
main( ) 함수를 종료해라!

```
#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return;
}
```

반환할 것이 없으니 main( )  
함수를 그냥 종료해라!

## 2.2 Hello C world

-교재 34페이지 -

## 2.2 Hello C world (1/2) - 소스코드

// 예제 2-1.c

```
01행 #include <stdio.h>
      int main(void)
      {
04행     printf("Hello C world");

06행     return 0;
      }
```

## 2.2 Hello C world (2/2) - 주석

### // 예제 2-1.c

```
01행 #include <stdio.h> // 전처리하는 부분
int main(void) // 운영체제가 호출
{
04행     printf("Hello C world"); // 모니터에 출력
06행     return 0; // 운영체제에게 값 반환 후, main() 종료
}
```

## 2.3 모니터에 데이터 출력하기 – printf() 함수

-교재 35페이지 -

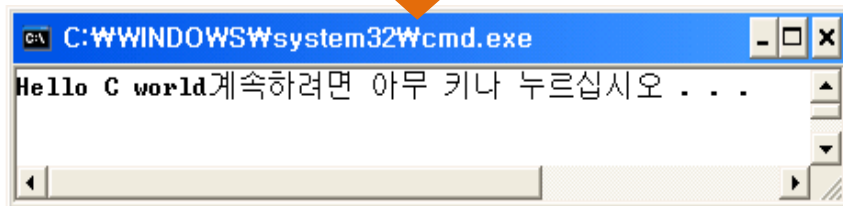
## 2.3 모니터에 데이터 출력하기 – printf() 함수 (1/18)

### ▶ 2-1.c 와 2-2.c 의 실행 결과 비교

// 예제 2-1.c

```
#include <stdio.h>
int main(void)
{
    printf("Hello C world");

    return 0;
}
```

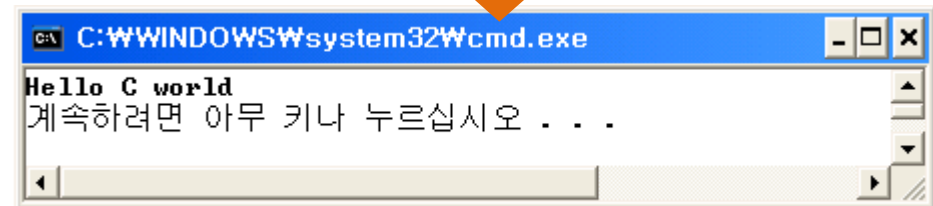


// 예제 2-2.c

```
#include <stdio.h>
int main(void)
{
    printf("Hello C world \u000A");

    return 0;
}
```

특수 문자  
↓  
\u000A



## 2.3 모니터에 데이터 출력하기 – printf() 함수 (2/18)

▶ **특수 문자:** 특수한 역할을 하는 문자

특수 문자	설명
\a	경고음 소리 발생
\b	백스페이스(Backspace)
\f	폼 피드(Form Feed)
\n	개행(New Line)
\r	캐리지 리턴(Carriage Return)
\t	수평 탭
\v	수직 탭
\w	역슬래시(\)
'\'	작은따옴표
\"	큰따옴표

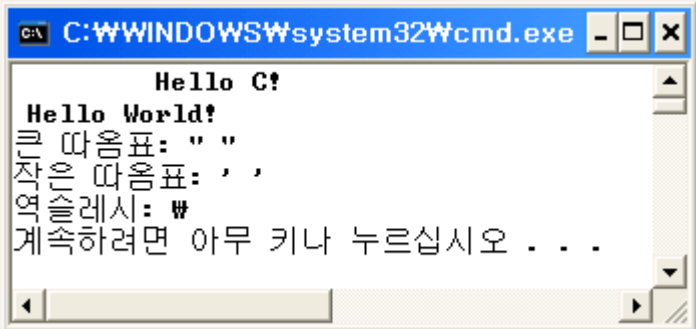


## 2.3 모니터에 데이터 출력하기 – printf() 함수 (3/18)

### ▶ 특수 문자 예제 실습 : 2-3.c

```
#include <stdio.h>
int main(void)
{
    printf("\t Hello C! \n Hello World! \n");
    printf("큰따옴표: \" \" \n");
    printf("작은따옴표: ' ' \n");
    printf("역슬레시: \\ \n");

    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
Hello C!
Hello World!
큰따옴표: " "
작은따옴표: ' '
역슬레시: \
계속하려면 아무 키나 누르십시오 . . .
```

## 2.3 모니터에 데이터 출력하기 – printf() 함수 (4/18)

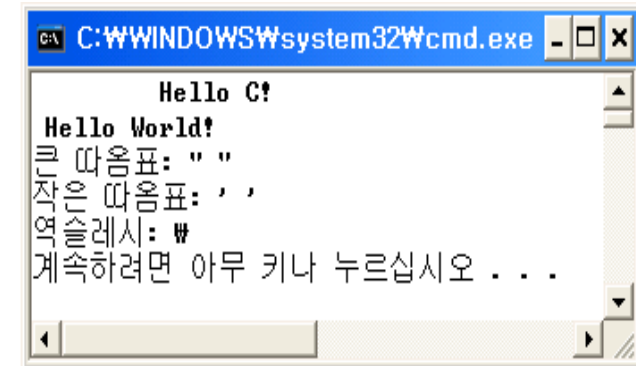
### ▶ 특수 문자 예제 분석 : 2-3.c

Tab 효과                      개행 문자                      개행 문자  
↓                                      ↓                                      ↓  
printf("Wt Hello C! Wn Hello world! Wn");

큰따옴표   큰따옴표   개행 문자  
↓                      ↓                      ↓  
printf("큰따옴표 : W" W" Wn");

작은따옴표   작은따옴표   개행 문자  
↓                      ↓                      ↓  
printf("작은따옴표 : W' W' Wn");

역슬래시   개행 문자  
↓                      ↓  
printf("역슬래시 : WW Wn");



## 2.3 모니터에 데이터 출력하기 - printf() 함수 (5/18)

### ▶ 서식 문자 (서식화된 문자)

- ✓ **printf() 함수** : Print에 Formatted에서 print에 f를 추가하여 만든 함수
- ✓ **Formatted의 의미** : '서식화된'의 의미
- ✓ **출력 서식 문자**

```
printf(" %d ", 3+5 );
```

3+5의 결과를 출력 서식 문자 %d로...

## 2.3 모니터에 데이터 출력하기 – printf() 함수 (6/18)

### ▶ 출력 서식 문자 (서식화된 출력 문자)

서식문자	출력 형태
%d, %i	10진수 정수 (양수와 음수 모두 표현 가능)
%x, %o	16진수 정수, 8진수 정수 (양수만 표현 가능)
%f, %lf	10진수 실수 (양수와 음수 모두 표현 가능)
%c	한 개의 문자
%s	문자열
%u	10진수 정수 (양수만 표현 가능)
%e	e 표기법에 의한 실수
%E	E 표기법에 의한 실수
%g	소수점 이하 자리 수에 따라 %f, %e 둘 중 하나를 선택
%G	소수점 이하 자리 수에 따라 %f, %E 둘 중 하나를 선택
%%	% 기호 출력

## 2.3 모니터에 데이터 출력하기 - printf() 함수 (7/18)

### ▶ 출력 서식 문자 (서식화된 출력 문자)

#### ① 숫자 출력하기 - %d, %i, %x, %o, %f, %lf

서식문자	출력 형태
%d, %i	10진수 정수 (양수와 음수 모두 표현 가능)
%x, %o	16진수 정수, 8진수 정수 (양수만 표현 가능)
%f, %lf	10진수 실수 (양수와 음수 모두 표현 가능)

## 2.3 모니터에 데이터 출력하기 – printf() 함수 (8/18)

### ▶ 숫자 출력하기 예제 실습 및 분석 : 2-4.c

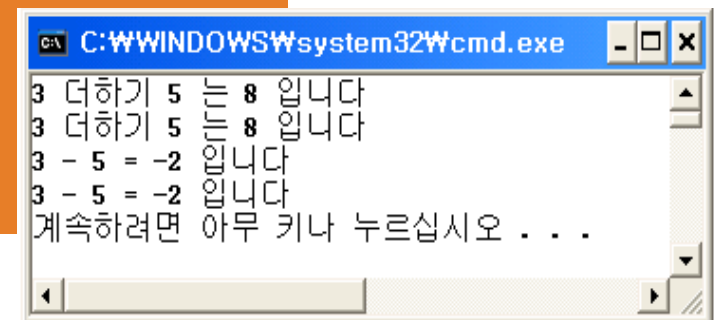
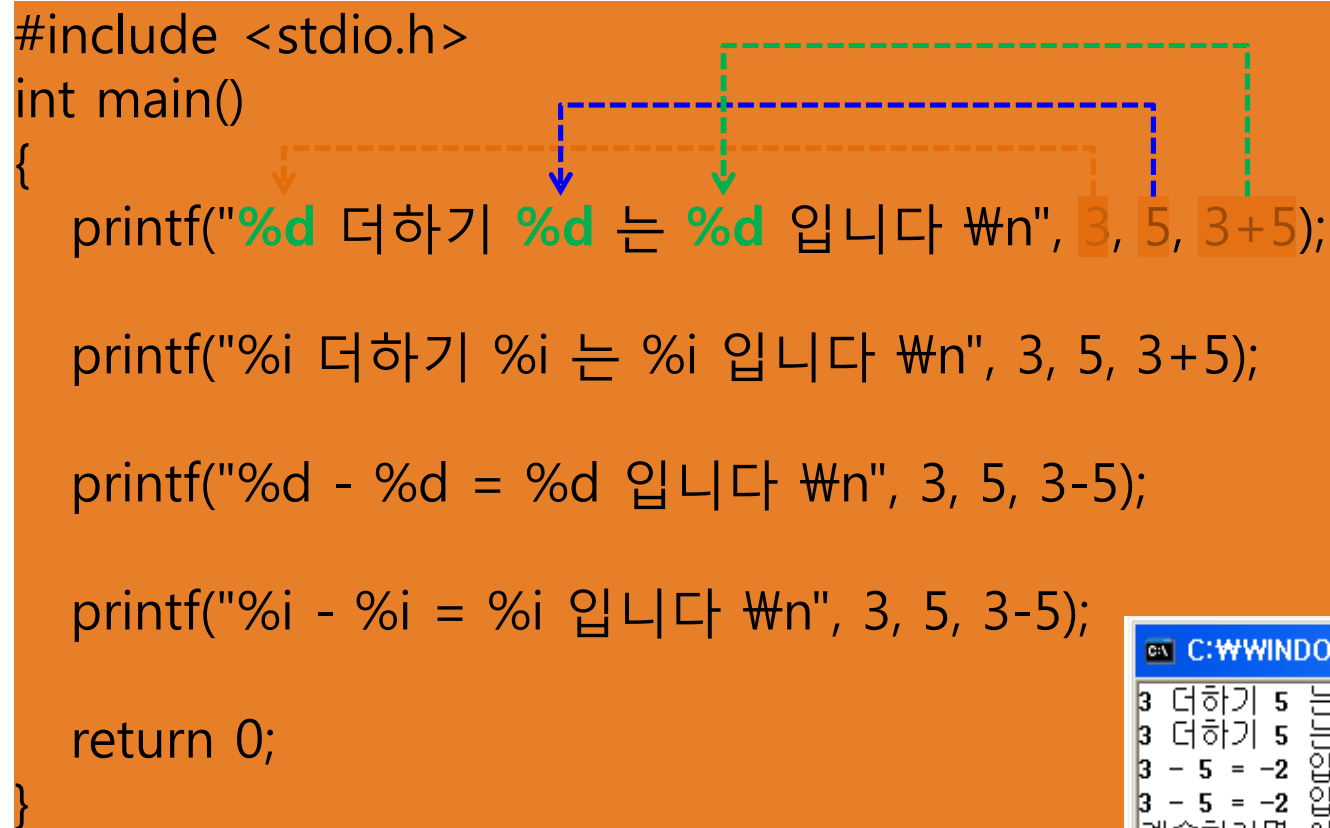
```
#include <stdio.h>
int main()
{
    printf("%d 더하기 %d 는 %d 입니다 \n", 3, 5, 3+5);

    printf("%i 더하기 %i 는 %i 입니다 \n", 3, 5, 3+5);

    printf("%d - %d = %d 입니다 \n", 3, 5, 3-5);

    printf("%i - %i = %i 입니다 \n", 3, 5, 3-5);

    return 0;
}
```



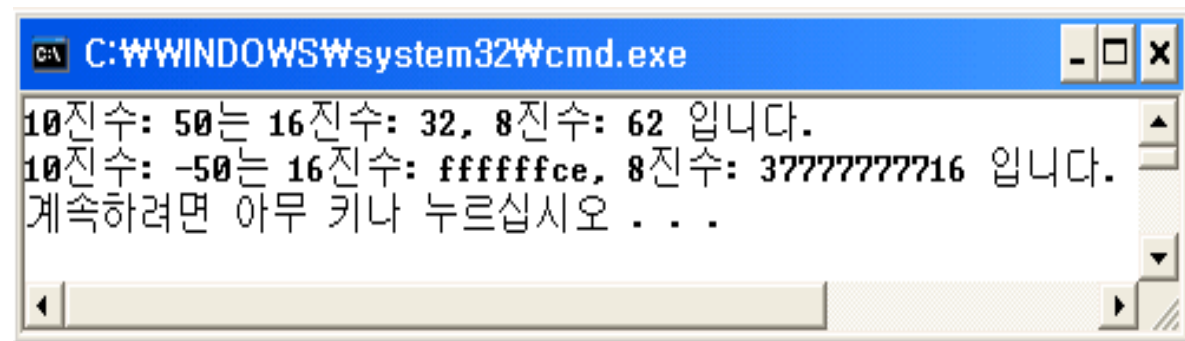
```
C:\WINDOWS\system32\cmd.exe
3 더하기 5 는 8 입니다
3 더하기 5 는 8 입니다
3 - 5 = -2 입니다
3 - 5 = -2 입니다
계속하려면 아무 키나 누르십시오 . . .
```

## 2.3 모니터에 데이터 출력하기 – printf() 함수 (9/18)

### ▶ 숫자 출력하기 예제 실습 : 2-5.c

```
#include <stdio.h>
int main()
{
    printf("10진수: %d는 16진수: %x, 8진수: %o 입니다. \n", 50, 50, 50);
    printf("10진수: %d는 16진수: %x, 8진수: %o 입니다. \n", -50, -50, -50);
    // 16진수와 8진수는 음수 표현 불가

    return 0;
}
```



C:\WINDOWS\system32\cmd.exe

```
10진수: 50는 16진수: 32, 8진수: 62 입니다.
10진수: -50는 16진수: fffffffce, 8진수: 3777777716 입니다.
계속하려면 아무 키나 누르십시오 . . .
```

## 2.3 모니터에 데이터 출력하기 - printf() 함수 (10/18)

### ▶ 숫자 출력하기 예제 분석 : 2-5.c

```
printf("10진수: %d는 16진수: %x, 8진수: %o 입니다. \n", 50, 50, 50);
```

```
printf("10진수: %d는 16진수: %x, 8진수: %o 입니다. \n", -50, -50, -50);
```

// 16진수와 8진수는 음수 표현 불가



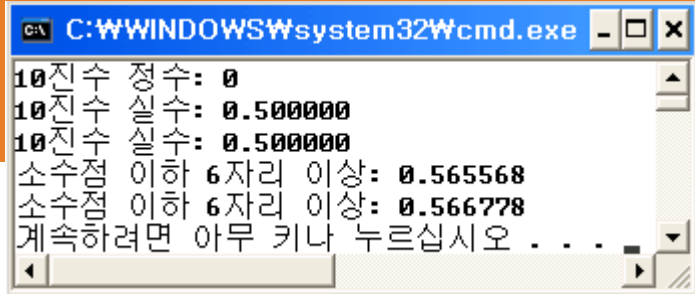
## 2.3 모니터에 데이터 출력하기 – printf() 함수 (11/18)

### ▶ 숫자 출력하기 예제 실습 : 2-6.c

```
#include <stdio.h>
int main(void)
{
    printf("10진수 정수: %d  \n", 0.5);
    printf("10진수 실수: %f  \n", 0.5);
    printf("10진수 실수: %lf \n", 0.5);

    printf("소수점 이하 6자리이상: %f  \n", 0.5655678);
    printf("소수점 이하 6자리이상: %lf \n", 0.5667784);

    return 0;
}
```



## 2.3 모니터에 데이터 출력하기 – printf() 함수 (12/18)

### ▶ 출력 서식 문자 (서식화된 출력 문자)

#### ② 문자와 문자열 출력하기- %c, %s

서식문자	출력 형태
%c	한 개의 문자
%s	문자열

## 2.3 모니터에 데이터 출력하기 – printf() 함수 (13/18)

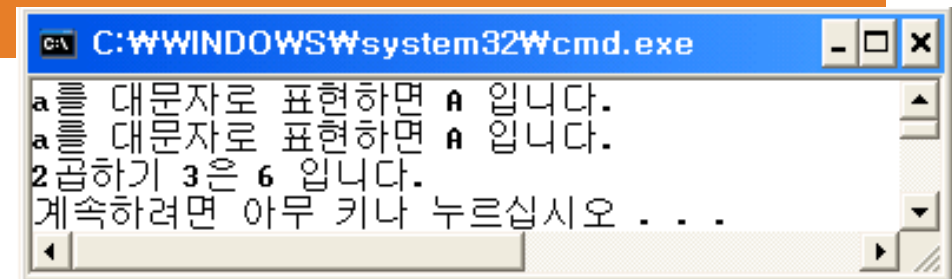
### ▶ 문자와 문자열 출력하기 예제 실습 : 2-7.c

```
#include <stdio.h>
int main(void)
{
    printf("a를 대문자로 표현하면 %c 입니다. \n", 'A');

    printf("%s %c 입니다. \n", "a를 대문자로 표현하면", 'A');

    printf("%s %d %s \n", "2곱하기3은", 2*3, "입니다.");

    return 0;
}
```



## 2.3 모니터에 데이터 출력하기 – printf() 함수 (14/18)

### ▶ 출력 서식 문자 (서식화된 출력 문자)

#### ③ 10진수 정수(양수) 출력하기- %u

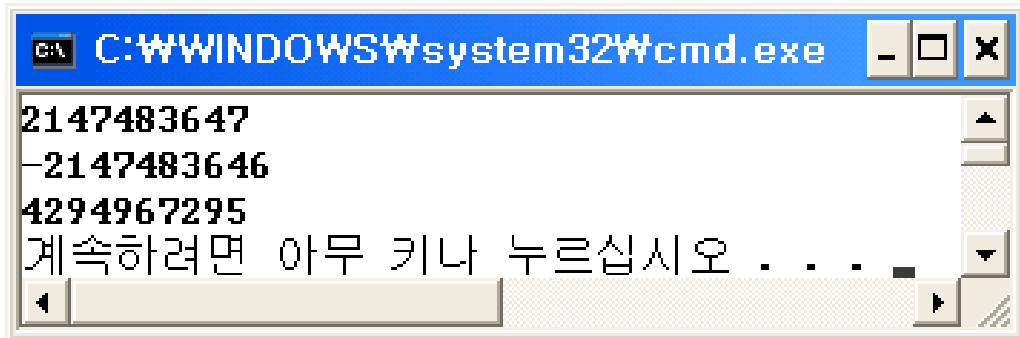
서식문자	출력 형태
%u	10진수 정수 (양수만 표현 가능)

## 2.3 모니터에 데이터 출력하기 - printf() 함수 (15/18)

### ▶ %u 관련 예제 실습 : 2-8.c

```
#include <stdio.h>
int main(void)
{
    printf("%d \n", 2147483647);    // 2147483647
    printf("%d \n", 2147483650);    // -2147483646
    printf("%u \n", 4294967295);    // 2147483647 의 2배

    return 0;
}
```



## 2.3 모니터에 데이터 출력하기 - printf() 함수 (16/18)

### ▶ 출력 서식 문자 (서식화된 출력 문자)

#### ④ 기타 실수 표기법 - %e, %E, %g, %G, %%

서식문자	출력 형태
%e	e 표기법에 의한 실수
%E	E 표기법에 의한 실수
%g	소수점 이하 자리 수에 따라 %f, %e 둘 중 하나를 선택
%G	소수점 이하 자리 수에 따라 %f, %E 둘 중 하나를 선택
%%	% 기호 출력

## 2.3 모니터에 데이터 출력하기 – printf() 함수 (17/18)

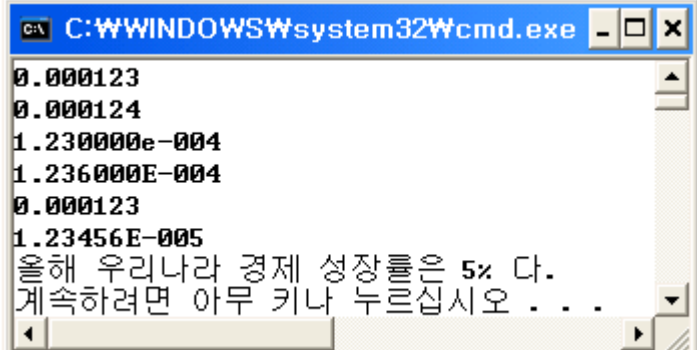
### ▶ 기타 실수 표기법 예제 실습 : 2-9.c

```
#include <stdio.h>
int main(void)
{
    printf("%f \n", 0.000123);           // 소수점 이하 6자리
    printf("%f \n", 0.0001236);         // 소수점 이하 6자리 초과

    printf("%e \n", 0.000123);           // 소수점 이하 6자리
    printf("%E \n", 0.0001236);         // 소수점 이하 6자리 초과

    printf("%g \n", 0.000123);           // 소수점 이하 6자리
    printf("%G \n", 0.0000123456);       // 소수점 이하 6자리 초과

    printf("올해 우리나라 경제 성장률은 5%% 다. \n");
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
0.000123
0.000124
1.230000e-004
1.236000E-004
0.000123
1.23456E-005
올해 우리나라 경제 성장률은 5% 다.
계속하려면 아무 키나 누르십시오 . . .
```

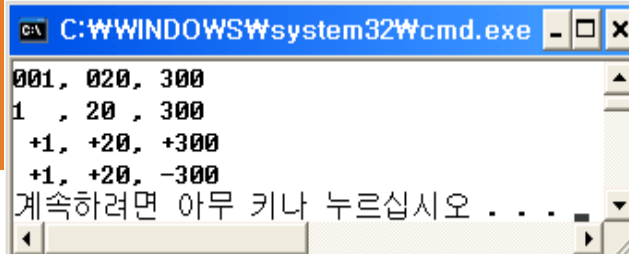
## 2.3 모니터에 데이터 출력하기 - printf() 함수 (18/18)

### ▶ 필드 폭 지정 (교재 46페이지)

- ✓ **%03d** : 필드 폭을 3칸 확보하고 오른쪽 정렬, 남은 자리는 0으로 채움
- ✓ **%-3d** : 필드 폭을 3칸 확보하고 왼쪽 정렬
- ✓ **%+3d** : 필드 폭 3칸 확보하고 오른쪽 정렬, 양수는 +, 음수는 - 출력

```
#include <stdio.h>
int main(void)
{
    printf("%03d, %03d, %03d \n", 1, 20, 300);
    printf("%-3d, %-3d, %-3d \n", 1, 20, 300);
    printf("%+3d, %+3d, %+3d \n", 1, 20, 300);
    printf("%+3d, %+3d, %+3d \n", 1, 20, -300);

    return 0;
}
```



C:\WINDOWS\system32\cmd.exe

```
001, 020, 300
1 , 20 , 300
+1, +20, +300
+1, +20, -300
계속하려면 아무 키나 누르십시오 . . .
```



## 2.4 키보드로 데이터 입력받기 – scanf() 함수

-교재 47페이지 -

## 2.4 키보드로부터 데이터 입력받기 – scanf() 함수 (1/8)

### ▶ printf()함수 vs. scanf()함수

#### ✓ printf()함수

- 모니터에 데이터를 출력하는 함수
- **stdio.h** 헤더 파일이 필요
- Print에 Formatted에서 **print**에 **f**를 추가하여 만든 함수 (출력 서식 필요)

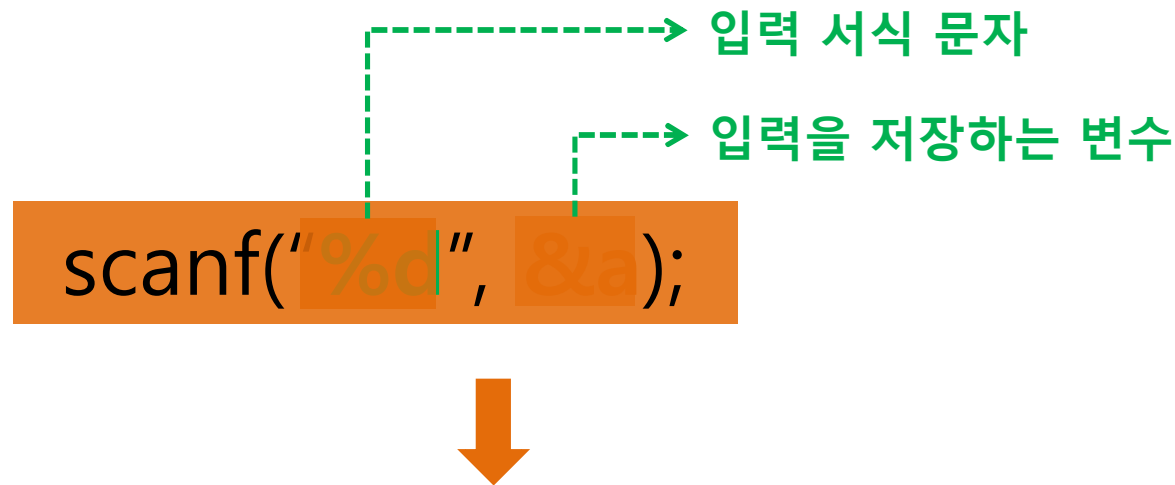
#### ✓ scanf()함수

- 키보드로 데이터를 입력하는 함수
- **stdio.h** 헤더 파일이 필요
- Scan에 Formatted에서 **scan**에 **f**를 추가하여 만든 함수 (입력 서식 필요)

## 2.4 키보드로부터 데이터 입력받기 – scanf() 함수 (2/8)

### ▶ scanf()함수의 기본 구조

- ✓ 입력 서식 문자
- ✓ 변수 (데이터를 저장하는 임시 공간)



#### <해석>

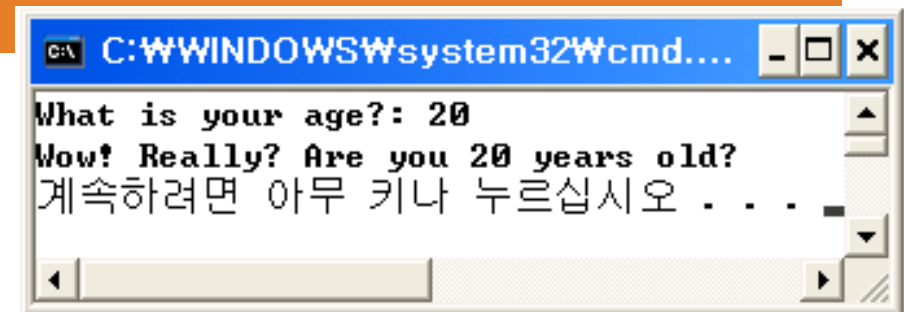
- ① 키보드로부터 데이터를 입력 서식 문자 `%d` 형식으로 입력 받는다.
- ② 입력받은 데이터를 변수 `a`에 저장한다.

## 2.4 키보드로부터 데이터 입력받기 – scanf() 함수 (3/8)

### ▶ scanf() 함수 예제 실습 : 2-11.c

```
#include <stdio.h>
int main(void)
{
    int age;
    printf("What is your age?: ");
    scanf("%d", &age);
    printf("Wow! Really? Are you %d years old?\n", age);

    return 0;
}
```



## 2.4 키보드로부터 데이터 입력받기 - scanf() 함수 (4/8)

### ▶ 입력 서식 문자 (서식화된 입력 문자)

서식문자	입력 형태
%d	10진수 정수 입력
%x	16진수 정수 입력
%o	8진수 정수 입력
%f	float형 실수 입력
%lf	double형 실수 입력
%c	한 개의 문자 입력 <나중에 배울 예정>
%s	문자열 입력 <나중에 배울 예정>
%u	10진수 정수(양수만 표현 가능) 입력 - <나중에 배울 예정>
%e	float형 e 표기법에 의한 실수 입력
%le	double형 e표기법에 의한 실수 입력

## 2.4 키보드로부터 데이터 입력받기 - scanf() 함수 (5/8)

### ▶ 입력 서식 문자 (서식화된 출력 문자)

#### ① 정수 입력 하기 - %d, %x, %o

서식문자	입력 형태
%d	10진수 정수 입력
%x	16진수 정수 입력
%o	8진수 정수 입력

## 2.4 키보드로부터 데이터 입력받기 – scanf() 함수 (6/8)

### ▶ scanf() 함수 예제 실습 : 2-12.c

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

```
int main(void)
```

```
{
```

```
    int a, b, c;
```

```
    printf("10진수 정수 1개 입력: ");
```

```
    scanf("%d", &a);
```

```
    printf("10진수: %d, 16진수: %x, 8진수: %o \\\n", a, a, a);
```

```
    printf("16진수 정수 1개 입력: ");
```

```
    scanf("%x", &b);
```

```
    printf("10진수: %d, 16진수: %x, 8진수: %o \\\n", b, b, b);
```

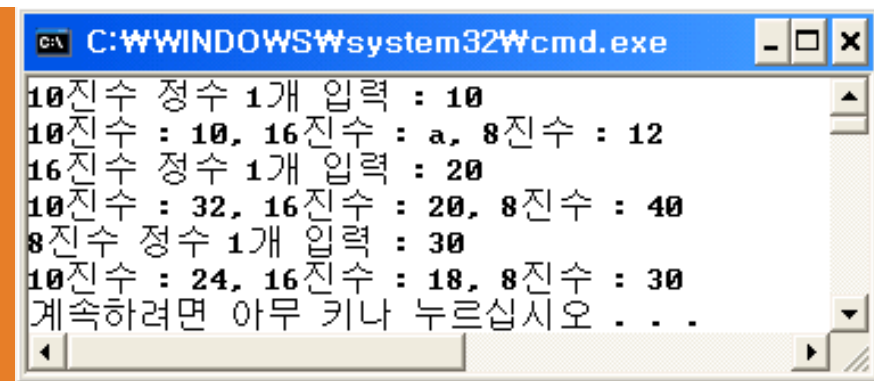
```
    printf("8진수 정수 1개 입력: ");
```

```
    scanf("%o", &c);
```

```
    printf("10진수: %d, 16진수: %x, 8진수: %o \\\n", c, c, c);
```

```
    return 0;
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
10진수 정수 1개 입력 : 10
10진수 : 10, 16진수 : a, 8진수 : 12
16진수 정수 1개 입력 : 20
10진수 : 32, 16진수 : 20, 8진수 : 40
8진수 정수 1개 입력 : 30
10진수 : 24, 16진수 : 18, 8진수 : 30
계속하려면 아무 키나 누르십시오 . . .
```

## 2.4 키보드로부터 데이터 입력받기 - scanf() 함수 (7/8)

### ▶ 입력 서식 문자 (서식화된 출력 문자)

#### ② 실수 입력 하기 - %f, %lf, %e, %le

서식문자	입력 형태
%f	float형 실수 입력
%lf	double형 실수 입력
%e	float형 e 표기법에 의한 실수 입력
%le	double형 e표기법에 의한 실수 입력



## 2.4 키보드로부터 데이터 입력받기 – scanf() 함수 (8/8)

### ▶ scanf() 함수 예제 실습 : 2-13.c

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

```
int main(void)
```

```
{
```

```
    float f1, f2;
```

```
    double d1, d2;
```

```
    printf("float형 실수 두 개 입력: ");
```

```
    scanf("%f %e", &f1, &f2);
```

```
    printf("float형 실수 출력: f1=%f, f2=%e \n", f1, f2);
```

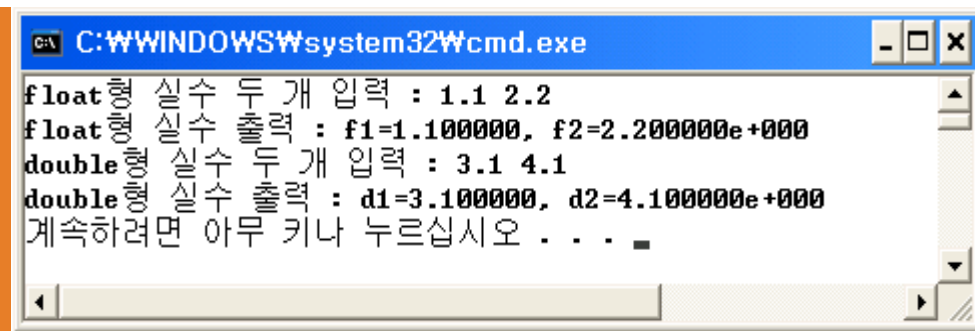
```
    printf("double형 실수 두 개 입력: ");
```

```
    scanf("%lf %le", &d1, &d2);
```

```
    printf("double형 실수 출력: d1=%lf, d2=%le \n", d1, d2);
```

```
    return 0;
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
float형 실수 두 개 입력 : 1.1 2.2
float형 실수 출력 : f1=1.100000, f2=2.200000e+000
double형 실수 두 개 입력 : 3.1 4.1
double형 실수 출력 : d1=3.100000, d2=4.100000e+000
계속하려면 아무 키나 누르십시오 . . .
```

## 공부한 내용 떠올리기

- ▶ C언어의 **기본 구조**
- ▶ **모니터**에 데이터를 **출력**하기 위해 **printf()함수**에 사용되는 서식 문자
- ▶ **키보드로** 데이터를 **입력**받기 위해 **scanf()함수**에 사용되는 서식 문자
- ▶ scanf() 함수로 입력받은 데이터를 저장하고 활용하는 방법

## 봉변 부른 부자의 교만 (출처: 사랑과 지혜의 탈무드 )

