

프로그래밍연습 Lab 1

# C 프로그램 구성요소

---

[TA] 강성민, 김기현, 최석원, 최지은, 표지원

Department of Computer Science and Engineering

Seoul National University, Korea

2022/09/14

# 이번 장에서 학습할 내용

- 주석, 들여쓰기
- 변수, 상수
- 산술연산과 대입 연산
- 출력함수와 입력 함수 (printf(), scanf())
- 형식 지정자

# 일반적인 프로그램의 형태

- 데이터를 받아서, 데이터를 처리한 후, 결과를 화면에 출력한다.

입력단계



처리단계



출력단계



# 덧셈 프로그램 (sum.c)

```
#include <stdio.h>
int main(void) { int x; int y; int sum;
x = 100; y = 200; sum = x + y;
printf("두 수의 합: %d", sum); return 0; }
```

```
/* 두 숫자의 합을 계산하는 프로그램*/
#include <stdio.h>

int main(void)
{
    int x;    //첫 번째 정수를 저장할 변수
    int y;    //두 번째 정수를 저장할 변수
    int sum;  //두 정수의 합을 저장하는 변수

    x = 100;
    y = 200;

    sum = x + y;
    printf("두 수의 합: %d", sum);

    return 0;
}
```

# 덧셈 프로그램 (sum\_1.c)

주석

```
/* 두 숫자의 합을 계산하는 프로그램*/
```

전처리기 지시어

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

```
int main(void)
```

```
{
```

```
int x;    //첫 번째 정수를 저장할 변수  
int y;    //두 번째 정수를 저장할 변수  
int sum;  //두 정수의 합을 저장하는 변수
```

변수 선언

함수

```
x = 100;
```

```
y = 200;
```

문장

```
sum = x + y;
```

연산

```
printf("두 수의 합: %d", sum);
```

```
return 0;
```

```
}
```

# 덧셈 프로그램 (sum\_1.c) 실행 결과

- 컴파일: `gcc sum_1.c -o sum_1`
- 실행 : `./sum_1`
- 두 수의 합: 300

# 주석(comment)

- 주석: 코드를 설명하는 글

- 컴파일러에 의해 무시됨
- 코드를 더 쉽게 이해할 수 있도록 함

```
1  /* 두 숫자의 합을 계산하는 프로그램*/
```

```
6      int x;    //첫 번째 정수를 저장할 변수
7      int y;    //두 번째 정수를 저장할 변수
8      int sum;  //두 정수의 합을 저장하는 변수
```

- 3가지 주석 작성 방법

- **/\*** 한 줄로 된 주석 **\*/**
- **/\*** -----

학번: 2022-27548

이름: 홍길동

여러 줄로 이루어진 주석

----- **\*/**

- **//** 여기서부터 줄 끝까지 주석

# 들여쓰기(indentation)

- 들여쓰기: 같은 수준에 있는 문장들을 왼쪽 끝에서 몇 자 안으로 들여쓰는 것

빈 줄을 넣어 의미별로 구별한다.

같은 내용의 처리이면 들여쓰기 한다.  
(tab 1 번 혹은 띄어쓰기 4번)

```
/* 두 숫자의 합을 계산하는 프로그램*/
#include <stdio.h>

int main(void)
{
    int x;    //첫 번째 정수를 저장할 변수
    int y;    //두 번째 정수를 저장할 변수
    int sum;  //두 정수의 합을 저장하는 변수

    x = 100;
    y = 200;

    sum = x + y;
    printf("두 수의 합: %d", sum);

    return 0;
}
```



# 전처리기(Preprocessor)

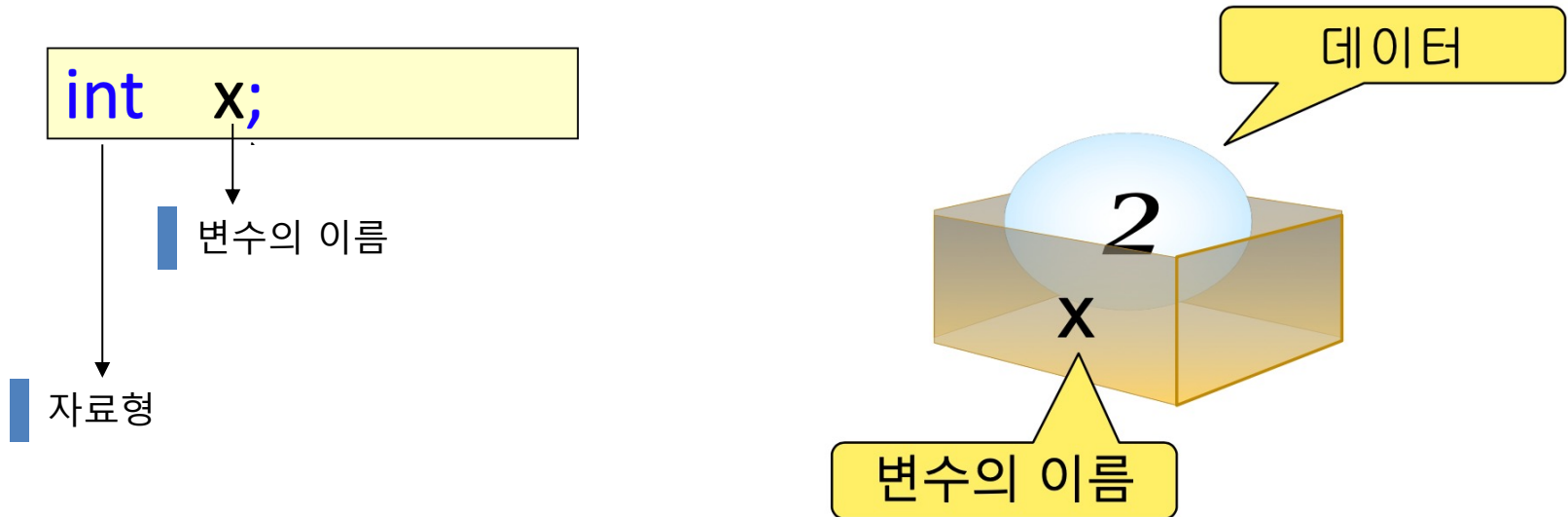
외부 파일을 포함시키라는 의미의 전처리기  
# 기호로 시작

**#include** <**stdio.h**>

Standard Input Output Header file  
표준 입출력에 대한 라이브러리 함수의 정의가 들어있음

# 변수(variable)

- 변수: 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간
- 데이터를 담는 상자

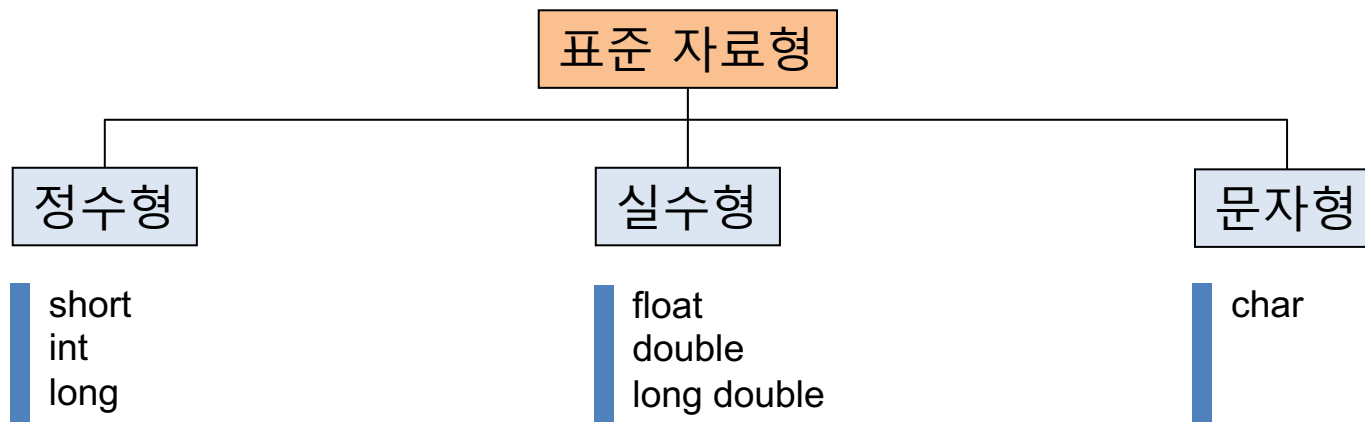


# 변수 (variable)

- 자료형(data type)

- 변수가 저장할 데이터가

정수인지 / 실수인지 / 또 다른 형태의 데이터인지 지정하는 것



# 변수 (variable)

- 식별자: 변수나 함수의 이름
- 식별자 만드는 규칙
  - 숫자로 시작 **No!**
  - C언어의 키워드와 똑같은 이름 **No!**
  - 영어 대소문자, 숫자, \_ **Yes!**

```
s  
s1  
student_number
```

올바른 식별자

```
^s  
2nd_student  
int
```

잘못된 식별자

# 변수 (variable)

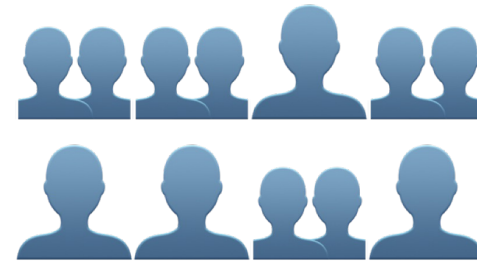
- 변수 선언

- 컴파일러에게 어떤 타입의 변수가 사용되는지 미리 알려주는 것

```
int x;  
int y;  
int sum;
```



지금부터 이 프로그램에서 사용될  
변수들을 소개하겠습니다.



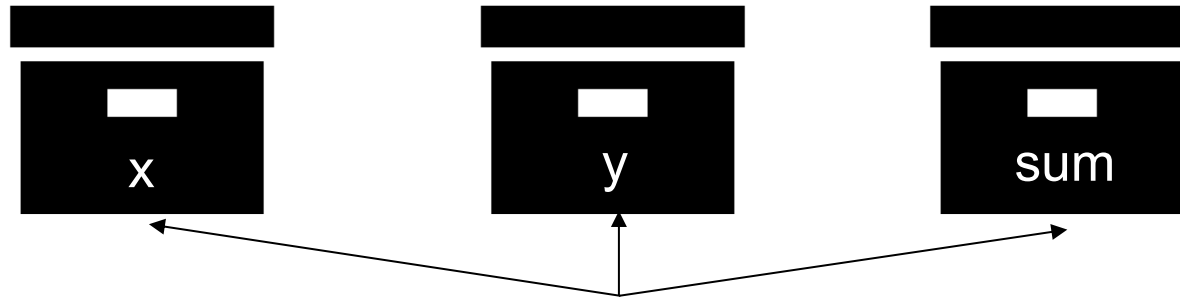
컴파일러

# 변수 (variable)

- 변수 선언

```
int x;    //첫 번째 정수를 저장할 변수  
int y;    //두 번째 정수를 저장할 변수  
int sum;  //두 정수의 합을 저장하는 변수
```

2) 각 변수는 정수를 저장할 수 있다.

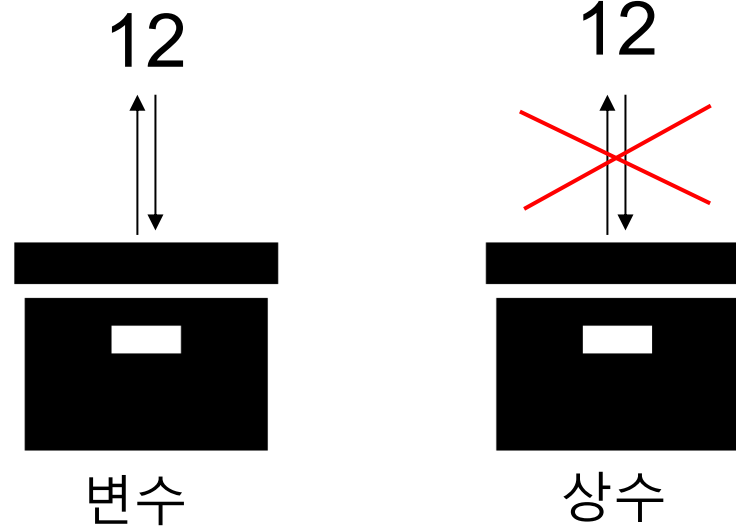


1) 메모리공간에 변수가 만들어지고 이름이 붙여진다

# 상수(constant)

## ■ 상수

- 그 값이 프로그램이 실행되는 동안 변하지 않는 수
- cf. 변수는 실행도중 값을 변경할 수 있지만, 상수는 한 번 정해지면 변경이 불가능



# 실습

- 1) int형 변수 i를 선언하는 문장을 작성해 보자.
- 2) double형 변수 f를 선언하는 문장을 작성해 보자.
- 3) 변수 선언의 위치는 함수의 어떤 위치에서 해야 하는가? (앞/뒤)



# 중간점검

- 1) int형 변수 i를 선언하는 문장을 작성해 보자. -> int i;
- 2) double형 변수 f를 선언하는 문장을 작성해 보자. -> double f;
- 3) 변수 선언의 위치는 함수의 어떤 위치에서 해야 하는가? (앞/뒤)

# 수식(expression)

- 수식
  - 피연산자(x, y)와 연산자(+)로 구성된 식
  - 수식은 결과값을 가진다.

```
sum = x + y;
```

# 산술연산과 대입연산

## ■ 산술연산

연산	연산자	C 수식	수학에서의 기호
덧셈	+	$x + y$	$x + y$
뺄셈	-	$x - y$	$x - y$
곱셈	*	$x * y$	$xy$
나눗셈	/	$x / y$	$x / y$
나머지	%	$x \% y$	$x \bmod y$

## ■ 대입연산(=할당연산=배정연산)

- 변수에 값을 저장하는 연산
- '=' 연산자는 변수에 값을 저장함

```
x = 100;  
y = 200;
```

# 정리

## ■ sum\_1.c

```
/* 두 숫자의 합을 계산하는 프로그램*/
#include <stdio.h>

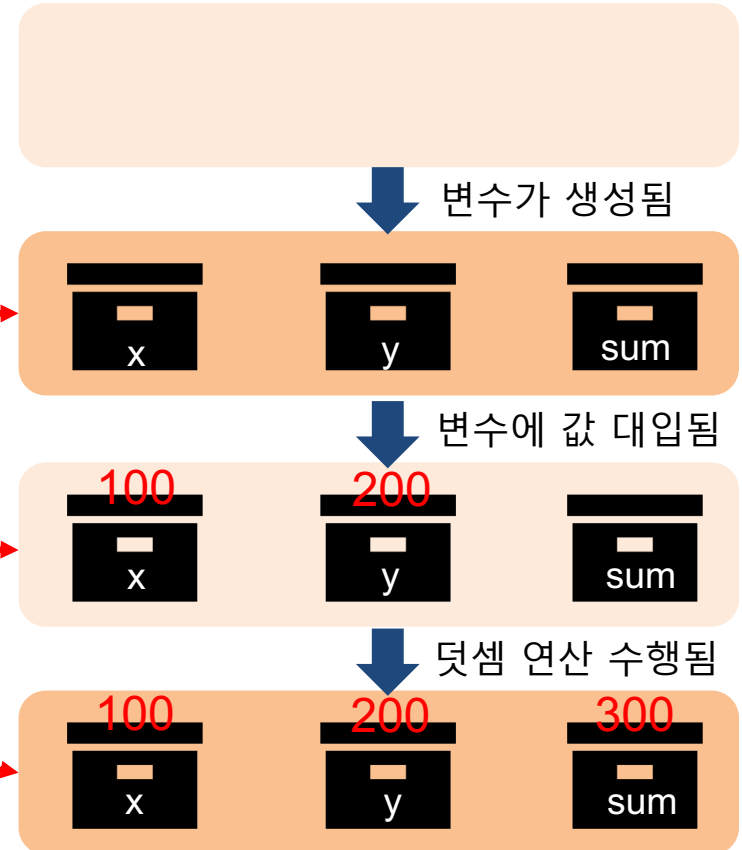
int main(void)
{
    int x; //첫 번째 정수를 저장할 변수
    int y; //두 번째 정수를 저장할 변수
    int sum; //두 정수의 합을 저장하는 변수

    x = 100;
    y = 200;

    sum = x + y;
    printf("두 수의 합: %d", sum);

    return 0;
}
```

## 컴퓨터 내부



# 실습(cal\_1.c)

- 문제

- $x$ 가 3일 때 수식  $x^2-5x+6$ 의 값을 계산하라.
- (HINT:  $x^2$ 는  $x*x$ 로 표현하고, 수식의 결과는  $y$ 로 나타내세요)

```
#include <stdio.h>

int main(void)
{

    printf("%d\n", y);

    return 0;
}
```

# 실습(cal\_1.c)

## ■ 정답 코드

1. cal\_1.c 파일 저장 후, git bash terminal에서 아래 단계 수행
2. 컴파일: gcc cal\_1.c -o cal\_1
3. 결과 출력: ./cal\_1

```
#include <stdio.h>

int main(void)
{
    int x;
    int y;

    x = 3;
    y = x*x-5*x+6;
    printf("%d\n",y);

    return 0;
}
```

# 실습(cal\_2.c)

## ■ 문제

- 변수 a와 변수 b의 곱을 변수 product에 저장하는 문장과,
- 변수 a를 변수 b로 나눈 값을 변수 quotient에 저장하는 문장을 작성 하여 보자.

```
/* 두 숫자를 곱하고 나누는 프로그램*/  
#include <stdio.h>  
  
int main(void)  
{  
    int a,b;        //두 정수를 저장할 변수를 선언  
    int product;    //두 정수의 곱을 저장하는 변수  
    int quotient;    //두 정수의 나눈 몫을 저장하는 변수  
  
    a = 100;  
    b = 2;  
  
    product  = a * b;  
    quotient = a / b;  
    printf("두 수의 곱: %d\n", product);  
    printf("두 수의 나눗셈: %d\n", quotient);  
  
    return 0;  
}
```

# 실습(cal\_2.c)

## ■ 정답 코드

1. cal\_2.c 파일 저장 후, git bash terminal에서 아래 단계 수행
2. 컴파일: gcc cal\_2.c -o cal\_2
3. 결과 출력: ./cal\_2

```
/* 두 숫자를 곱하고 나누는 프로그램*/
#include <stdio.h>

int main(void)
{
    int a,b;        //두 정수를 저장할 변수를 선언
    int product;    //두 정수의 곱을 저장하는 변수
    int quotient;   //두 정수의 나눈 몫을 저장하는 변수

    a = 100;
    b = 2;

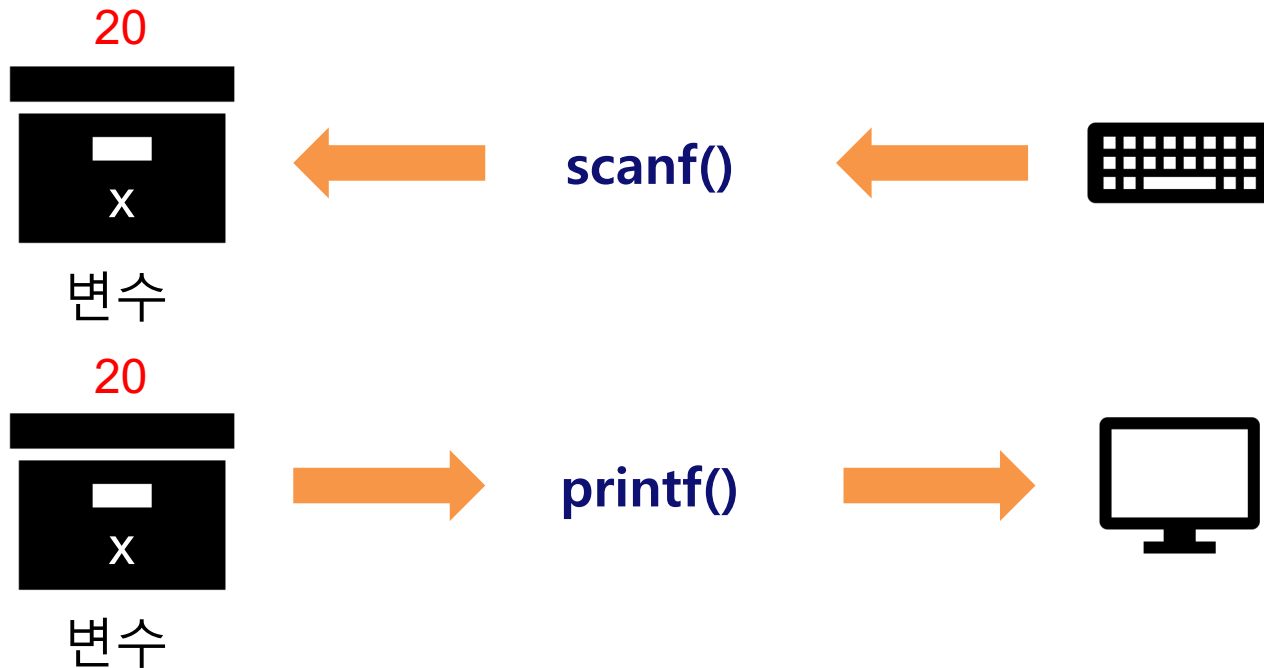
    product  = a * b;
    quotient = a / b;
    printf("두 수의 곱: %d\n", product);
    printf("두 수의 나눗셈: %d\n", quotient);

    return 0;
}
```



# 출력함수와 입력함수

- **printf()**
  - 모니터에 출력을 하기 위한 표준 출력 라이브러리
- **scanf()**
  - 표준 입력 함수. 사용자로부터 다양한 데이터를 서식에 맞춰 입력받을 수 있음



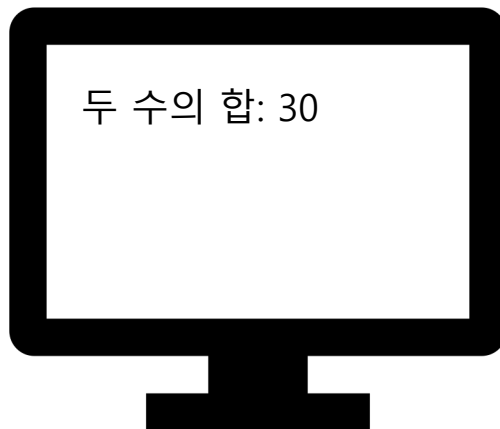
# printf()

- 문자열 출력

- `printf("Hello World!\n");`

- 변수값 출력

- `printf("두 수의 합: %d", sum);` → 형식 지정자와 변수의 개수/순서는 동일해야 함



# 형식 지정자

## ■ 형식 지정자

- printf()에서 값을 출력하는 형식을 지정

형식 지정자	의미	예	실행 결과
<b>%d</b>	10진 정수로 출력	printf("%d \n", 10);	10
<b>%f</b>	실수로 출력	printf("%f \n", 3.14);	3.14
<b>%c</b>	문자로 출력	printf("%c \n", 'a');	a
<b>%s</b>	문자열로 출력	printf("%s \n", "Hello");	Hello

### EX)

- printf("%d %f", number, grade);  
-> 23 3.99
- printf("%f \n", sum); //만약 sum이 정수형 변수라면?  
-> 형식 지정자와 변수의 타입이 다르므로 에러 발생

# 실습(sum\_2.c)

- 사용자로부터 두 정수를 scanf()로 입력 받아 합을 계산해 출력해 보자.

```
/* 사용자로부터 입력 받은 두 정수의 합을 계산하여 출력 */
#include <stdio.h>
int main(void)
{
    int x;
    int y;
    int sum;

    printf("첫번째 숫자를 입력하시오:");
    scanf("%d", &x);

    printf("두번째 숫자를 입력하시오:");
    scanf("%d", &y);

    sum = x + y;
    printf("두수의 합: %d \n", sum);

    return 0;
}
```

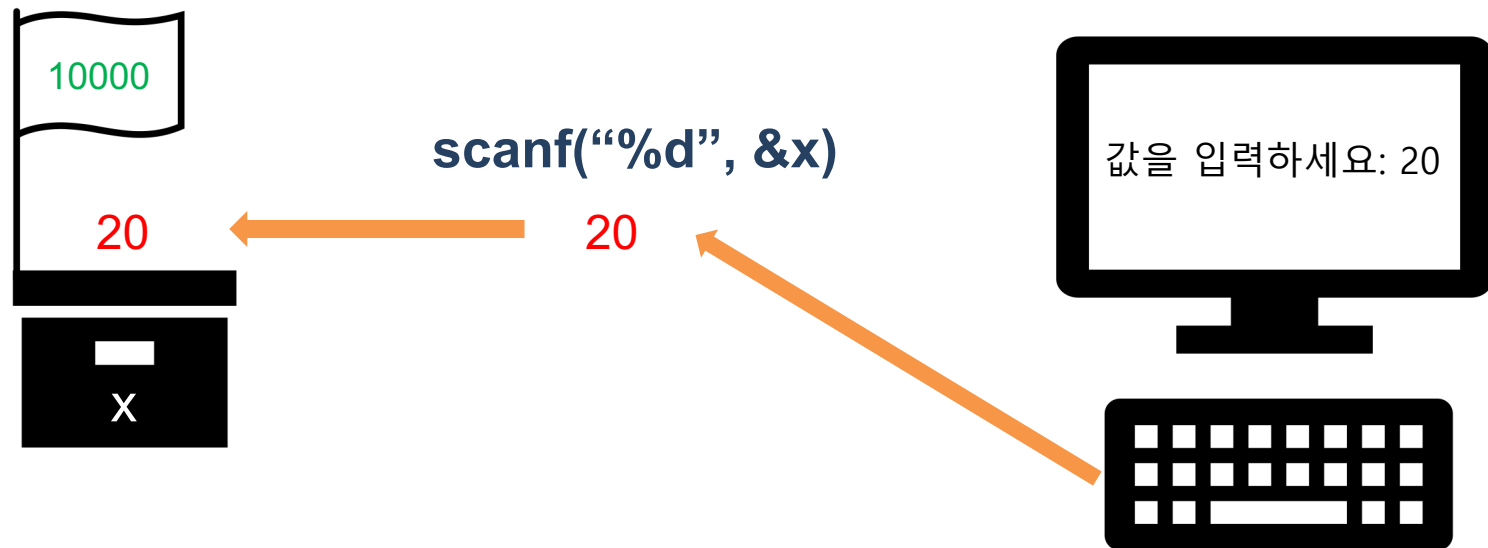
→ 하나의 정수를 입력받아 x라는 정수형 변수에 저장

```
첫 번째 숫자를 입력하시오 :10
두 번째 숫자를 입력하시오 :20
두 수의 합 : 30
```

# scanf()

## ■ scanf()의 동작

- 키보드로부터 값을 받아 변수에 저장
- 값을 저장하기 위해서는 변수의 주소를 알아야 함
- &(ampersand)는 해당 변수의 주소를 가리키는 역할



EX)

```
scanf("%d %f", &number, &grade);  
> 23 3.99
```

# 형식 지정자

형식 지정자	의미	예
%d	정수를 10진수로 입력한다	scanf("%d", &i);
%f	float 형의 실수로 입력한다.	scanf("%f", &f);
%lf	double 형의 실수로 입력한다.	scanf("%lf", &d);
%c	문자 형태로 입력한다.	scanf("%c", &ch);
%s	문자열 형태로 입력한다.	char s[10]; scanf("%s", &s);

# 실습(salary.c)

- `printf()`와 `scanf()`를 이용하여, 월급을 입력받아 10년간의 저축액을 구해보자
  - HINT
    - 1) 월급과 저축액을 정수형(int) 변수로 선언하자.
  - Output

```
월급을 입력하시오 (단위 : 만 원 ) : 200
10년 동안의 저축액 : 24000
```

# 실습(salary.c)

- 코드

```
/* 저축액을 계산하는 프로그램 */
#include <stdio.h>
int main(void){
    int salary; // 월급
    int deposit; // 저축액

    printf("월급을 입력하시오(단위: 만 원): ");
    scanf("%d", &salary); //사용자로부터 월급을 입력받는다.

    deposit = 10 * 12 * salary; //월급에 10*12를 곱하여 10년동안의 저축액을 계산한다.

    printf("10년 동안의 저축액: %d\n", deposit); //결과를 출력한다.

    return 0;

}
```



# 실습(circle.c)

- `printf()`와 `scanf()`를 이용하여 원의 면적을 계산하는 프로그램을 작성해 보자.

- HINT

- 1) 원의 반지름과 면적을 `float` 형의 실수 변수로 선언하자.
- 2)  $A = \pi r^2$  (cf.  $\pi$  는 3.14로 가정)

- Output

```
반지름을 입력하세요 : 4  
원의 면적 : 50.240002
```

# 실습(circle.c)

- 코드

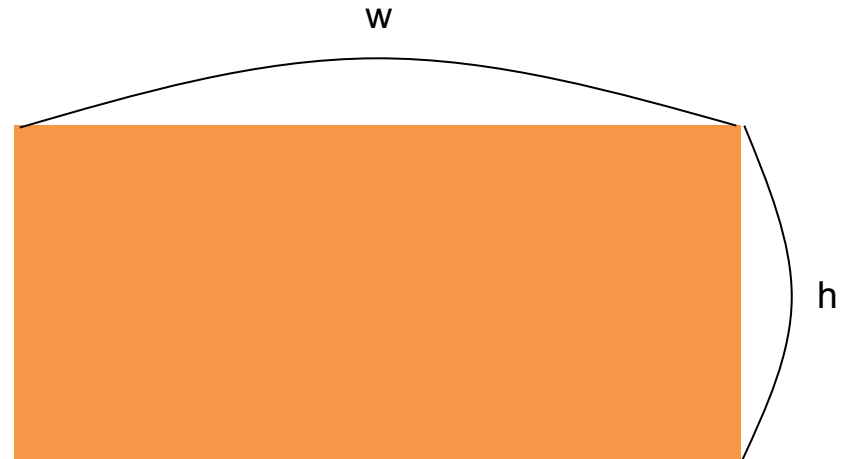
```
/* 원의 면적을 계산하는 프로그램*/  
#include <stdio.h>  
  
int main(void)  
{  
    float radius; // 원의 반지름  
    float area; // 면적  
  
    printf("반지름을 입력하세요: ");  
    scanf("%f", &radius);  
  
    printf("원의 면적: %f\n", area);  
  
    return 0;  
}
```

# 과제 안내

- 과제 게시판이 활성화되었음.
- 다음주 실습 수업 9/21(수) 1시간 전인 17:30까지 제출
- 공지 게시판 참고하여, 과제 게시판에 과제 제출

# 추가 실습(rectangle.c)

- `printf()`를 이용하여 사각형의 둘레와 면적을 계산하는 프로그램을 작성해 보자.
  - HINT
    - 1) 필요한 변수는 `w`, `h`, `area`, `perimeter`라고 하자.
    - 2) 변수의 자료형은 실수를 저장할 수 있는 `double`형으로 하자.
    - 3) `area = w*h;`
    - 4) `perimeter = 2*(w+h);`



# 추가 실습(rectangle.c)

- 코드

```
#include <stdio.h>

int main(void) {
    double w;
    double h;
    double area;
    double perimeter;

    w = 10.0;
    h = 5.0;
    area = w*h;
    perimeter = 2*(w+h);

    printf("사각형의 넓이: %lf", area);
    printf("사각형의 둘레: %lf", perimeter);
    return 0;
}
```

# 추가 실습(rectangle.c)

## ■ High Level 도전 문제

1. 변수들을 한 줄에 모두 선언해 보자
2. 한 번의 printf() 호출로 area, perimeter를 동시에 출력해 보자.
3. w와 h의 값을 사용자로부터 받도록 변경해 보자.(%lf 사용)

## ■ Output

```
width:10.0  
height:5.0  
사각형의 넓이 : 50.000000  
사각형의 둘레 : 30.000000
```

- 변경된 코드 정답은 ETL과 홈페이지에 15일에 올라갑니다.