

컴퓨터 프로그래밍 및 실습

3주차. 수식과 연산자

1. 수식과 연산자

■ 실습 제출 안내

- 솔루션 이름은 "Practice week 3"
- 프로젝트 이름과 소스코드 이름은 Problem1, Problem2, ...
 - 실습1의 프로젝트 이름은 Problem1, 소스코드 이름은 problem1.c
 - 실습 2의 프로젝트 이름은 Problem2, 소스코드 이름은 problem2.c ...
- 솔루션 폴더를 압축하여 Practice_week3_학번_이름.zip 으로 제출
- 제출기한: 당일 19시 까지

1. 수식과 연산자

■ 연산자 종류

연산자의 분류	연산자	의미
대입	=	오른쪽을 왼쪽에 대입
산술	+ - * / %	사칙연산과 나머지 연산
부호	+ -	양수와 음수 표시
증감	++ --	증가, 감소 연산
관계	> < == != >= <=	오른쪽과 왼쪽을 비교
논리	&& !	논리적인 AND, OR
조건	?	조건에 따라 선택
coma	,	피연산자들을 순차적으로 실행
비트 연산자	& ^ ~ << >>	비트별 AND, OR, XOR, 이동, 반전
sizeof 연산자	sizeof	자료형이나 변수의 크기를 바이트 단위로 반환
형변환	(type)	변수나 상수의 자료형을 변환
포인터 연산자	* & []	주소계산, 포인터가 가리키는 곳의 내용 추출
구조체 연산자	. ->	구조체의 멤버 참조

1. 수식과 연산자

■ 연산자 종류

- ❖ 산술 연산: 컴퓨터의 가장 기본적인 연산
- ❖ 덧셈, 뺄셈, 곱셈, 나눗셈 등의 사칙 연산을 수행하는 연산자

연산자	기호	사용예	결과값
덧셈	+	$7 + 4$	11
뺄셈	-	$7 - 4$	3
곱셈	*	$7 * 4$	28
나눗셈	/	$7 / 4$	1
나머지	%	$7 \% 4$	3

$$y = mx + b \quad \rightarrow y = m * x + b;$$

$$y = ax^2 + bx + c \quad \rightarrow y = a * x * x + b * x + c;$$

$$m = \frac{x + y + z}{3} \quad \rightarrow m = (x + y + z) / 3;$$

1. 수식과 연산자

■ 실습 1

- 1. int 타입의 변수 x와 y를 선언한다
- 2. x와 y를 scanf를 통해 값을 받아온다
- 3. 아래 수식을 구현한다

- $a = x + y \times \frac{10}{9} + \frac{3}{5}$

- $b = \frac{2x+3y+20}{10x+y} + \frac{1}{y}$

- $c = (x + y) \div b$

- $d = \begin{cases} 1, & \text{if } x \text{ is odd} \\ 0, & \text{if } x \text{ is even} \end{cases}$

- $e = \frac{x}{y}$

- 4. a, b, c, d, e를 출력한다.

1. 수식과 연산자

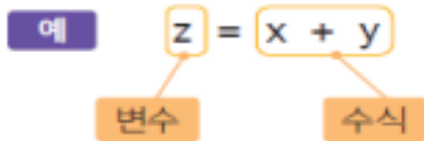
■ 증감 연산자

증감 연산자	차이점
$++x$	수식의 값은 증가된 x 값이다.
$x++$	수식의 값은 증가되지 않은 원래의 x 값이다.
$--x$	수식의 값은 감소된 x 값이다.
$x--$	수식의 값은 감소되지 않은 원래의 x 값이다.

1. 수식과 연산자

■ 복합 대입 연산자

복합 대입 연산자	의미	복합 대입 연산자	의미
$x += y$	$x = x + y$	$x \&= y$	$x = x \& y$
$x -= y$	$x = x - y$	$x = y$	$x = x y$
$x *= y$	$x = x * y$	$x ^= y$	$x = x ^ y$
$x /= y$	$x = x / y$	$x >>= y$	$x = x >> y$
$x \% = y$	$x = x \% y$	$x <<= y$	$x = x << y$



$x = x + 1;$

$y = x = 3;$

1. 수식과 연산자

■ 비교 연산자

연산	의미	연산	의미
$x == y$	x와 y가 같은가?	$x < y$	x가 y보다 작은가?
$x != y$	x와 y가 다른가?	$x >= y$	x가 y보다 크거나 같은가?
$x > y$	x가 y보다 큰가?	$x <= y$	x가 y보다 작거나 같은가?

1. 수식과 연산자

■ 논리 연산자

연산	의미
$x \ \&\& \ y$	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
$x \ \ y$	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
$!x$	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

1. 수식과 연산자

■ 논리 값

- C언어에서는 1을 true로, 0을 false로 처리한다.
- 따라서 $a > b$ 의 결과값은 0
- $a < b$ 의 결과값은 1

```
#include <stdio.h>

int main(void) {
    double a = 2.7;
    double b = 5.8;

    int result1 = a > b;
    int result2 = a < b;

    printf("%d %d\n", result1, result2);
    // 0 1 을 출력
}
```

1. 수식과 연산자

■ 논리 값

- 논리 값(true, false)를 int로 관리하면 헷갈리기도 하고, 가독성에 좋지 않다.
- stdbool.h에 boolean을 담을 수 있는 bool 자료형이 있다.
- true, false 값도 사용 가능

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

int main(void) {
    double a = 2.7;
    double b = 5.8;

    bool result1 = a > b;
    bool result2 = a < b;

    bool test = true && false || result1;

    printf("%d %d %d\n", result1, result2, test);
    // 0 1 을 출력
}
```

```
//
// stdbool.h
//
//      Copyright (c) Microsoft Corporation. All rights reserved.
//
// The C Standard Library <stdbool.h> header.
//
#ifndef _STDBOOL
#define _STDBOOL

#define __bool_true_false_are_defined 1

#ifndef __cplusplus
#define bool _Bool
#define false 0
#define true 1
#endif /* __cplusplus */

#endif /* _STDBOOL */
```

1. 수식과 연산자

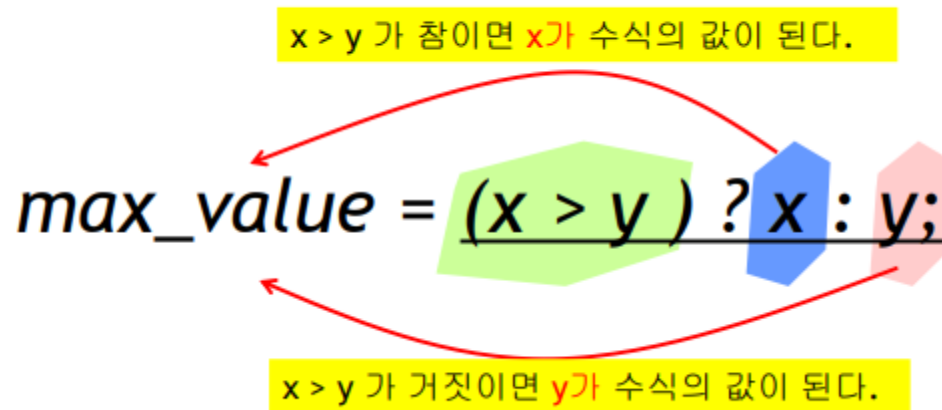
■ 실습 2

- 아래 조건에 맞게 참, 거짓 값을 구하는 프로그램을 개발해보자

- 1. 나이가 15세 이상 23세 미만이고 이름 첫 글자가 'K' 인가?
 - 필요 변수 → age(나이): int, f_name(이름 첫 글자): char
- 2. 키가 170이상, 180이하 이거나 학년이 18 이하인가?
 - 필요 변수 → height(키): double, adm_year(학번): int
- 3. 나이가 20세 이상이고 통금시간이 22시 이상인가? 만약 통금시간이 22시 미만이라면 부모님 동의를 받아야 한다.
 - 필요 변수 → age(나이): int, curfew(통금시간): int, is_approved(부모님 동의 유무): int[0|1]
- 4. 토익 점수가 750점 이상이고 정보처리기사 자격증을 취득해야 졸업할 수 있다. 만약 토익 점수가 750점 미만이라면 텡스 점수가 615점 이상이거나 토익 스피킹 점수가 6이상이거나 토플 점수가 557이상 이어야 한다. 졸업을 할 수 있는가?
 - 필요 변수 → toeic(토익점수): int, has_EIP(정보처리기사 취득 여부): int[0|1]
teps(텡스점수): int, ts(토익스피킹점수): int, toefl(토플점수): int

1. 수식과 연산자

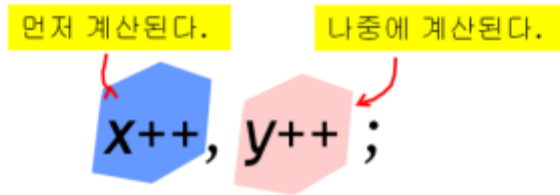
■ 조건 연산자



1. 수식과 연산자

■ 콤마 연산자

- ❖ 콤마로 연결된 수식은 순차적으로 계산
- ❖ 반복문이나 조건문에서 요긴하게 사용됨



1. 수식과 연산자

■ 비트 연산자

연산자	연산자의 의미	예
&	비트 AND	두개의 피연산자의 해당 비트가 모두 1이면 1, 아니면 0
	비트 OR	두개의 피연산자의 해당 비트중 하나만 1이면 1, 아니면 0
^	비트 XOR	두개의 피연산자의 해당 비트의 값이 같으면 0, 아니면 1
<<	왼쪽으로 이동	지정된 개수만큼 모든 비트를 왼쪽으로 이동한다.
>>	오른쪽으로 이동	지정된 개수만큼 모든 비트를 오른쪽으로 이동한다.
~	비트 NOT	0은 1로 만들고 1은 0로 만든다.

❖ 비트 AND 연산자

변수1	00000000	00000000	00000000	00001001	(9)
변수2	00000000	00000000	00000000	00001010	(10)
<hr/>					
(변수1 AND 변수2)	00000000	00000000	00000000	00001000	(8)

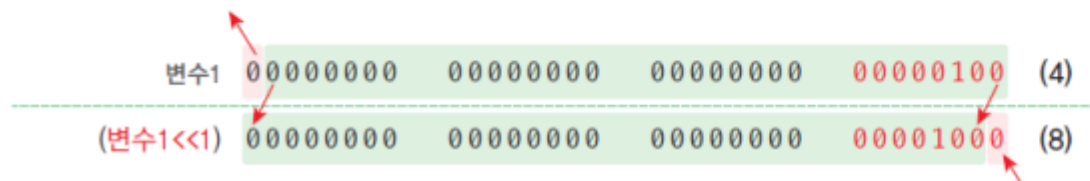
1. 수식과 연산자

■ 비트 쉬프트 연산자

연산자	기호	설명
왼쪽 비트 이동	\ll	$x \ll y$ x의 비트들을 y 칸만큼 왼쪽으로 이동
오른쪽 비트 이동	\gg	$x \gg y$ x의 비트들을 y 칸만큼 오른쪽으로 이동

\ll 연산자

- 비트를 왼쪽으로 이동
- 변수의 경계를 벗어나는 비트들은 없어지고 오른쪽 빈 공간은 0으로 채움
- 값은 2배가 됨



1. 수식과 연산자

■ 실습 3

- 달력은 지구가 태양을 공전하는 시간을 기준으로 작성됨.
- 지구가 태양을 한바퀴 도는데 걸리는 시간은 365일보다 1/4일 만큼 더 걸린다.
- 따라서 매 4년 마다 하루 정도 오차가 발생한다.
- 이것을 조정하기 위해서 윤년이 생겼다.
- 입력된 연도가 윤년인지를 판단하는 프로그램을 개발한다.
- 윤년이 되기 위한 조건
 - 연도가 4로 나누어 떨어진다.
 - 100으로 나누어 떨어지는 연도는 제외한다.
 - 400으로 나누어 떨어지는 연도는 포함한다.

