

3강

# C\_PROGRAMMING



# 산술연산자

❖ 두 개의 피 연산자 간의 산술연산을 하기 위해 사용

산술 연산자	사용예	의 미
+	$a+b$	두수의 합
-	$a-b$	두수의 차
*	$a*b$	두수의 곱
/	$a/b$	나누기 몫
%	$a\%b$	나누기 나머지

❖ %의 경우 짝,홀수 구분과 배수 구분 시 사용

## '%' 연산의 사용예

### ❖ 짝, 홀수 구분

- $10 \% 2 \Rightarrow 0$ 으로 짝수
- $15 \% 2 \Rightarrow 1$ 로 홀수

### ❖ 배수 구분

- $123 \% 3 \Rightarrow 0$ 으로 3의 배수

### ❖ 숫자의 자리수 구분하기

- $156 \% 10 \Rightarrow 6$
- $156 / 10 \Rightarrow 15$
- $15 \% 10 \Rightarrow 5$
- $15 / 10 \Rightarrow 1$

### ❖ 범위 안의 숫자 구하기

- $N \% 100 \Rightarrow 0 \sim 99$  사이의 값을 구할 수 있다

## 예제

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

```
main()
```

```
{
```

```
    int su1 = 20, su2 = 3;
```

```
    printf("%d + %d = %d\n", su1, su2, su1 + su2);
```

```
    printf("%d - %d = %d\n", su1, su2, su1 - su2);
```

```
    printf("%d * %d = %d\n", su1, su2, su1 * su2);
```

```
    printf("%d / %d = %d\n", su1, su2, su1 / su2);
```

```
    printf("%d %% %d = %d\n", su1, su2, su1 % su2);
```

```
}
```

## 관계연산자

- ❖ 두 개의 피 연산자 간의 대소관계를 비교하기 위하여 사용
- ❖ 0은 거짓 0이외의 모든 수는 참을 의미

연산자	의 미	사 용 예
<	...보다 작다	if(a<10)~
>	...보다 크다	if(a>10)~
<=	...보다 작거나 같다	if(a<=10)~
>=	...보다 크거나 같다	if(a>=10)~
==	...와 같다	if(a==10)~
!=	...와 같지 않다	if(a!=10)~

## 예제

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

```
main()
```

```
{
```

```
    float su1 = 3.01, su2 = 3.0;
```

```
    printf("변수 su1과 su2의 크기 비교 결과 : %d\\n", su1 <= su2);
```

```
    printf("변수 su1과 su2의 크기 비교 결과 : %d\\n", su1 >= su2);
```

```
    printf("변수 su1과 su2의 크기 비교 결과 : %d\\n", su1 == su2);
```

```
    printf("변수 su1과 su2의 크기 비교 결과 : %d\\n", su1 != su2);
```

```
}
```

# 대입연산자

## ❖ 정의

- 우측에 수행한 결과를 좌측에 지정된 변수로 대입

## ❖ 복합대입연산자

- 대입연산자를 다른 연산자와 결합하여 사용

복합 대입 연산자	사용예	의 미
$+=$	$a+=b$	$a=a+b$
$-=$	$a-=b$	$a=a-b$
$*=$	$a*=b$	$a=a*b$
$/=$	$a/=b$	$a=a/b$
$\%=$	$a\%=b$	$a=a\%b$

# 예제

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

```
main()
```

```
{
```

```
    int su1, su2;
```

```
    su1 = su2 = 5;
```

```
    printf("su1 + 1 = %d\n", su1 += 1);
```

```
    printf("su1 - 1 = %d\n", su1 -= 1);
```

```
    printf("su1 * su2 = %d\n", su1 *= su2);
```

```
    printf("su1 / su2 = %d\n", su1 /= su2);
```

```
    printf("su1 %% su2 = %d\n", su1 %= su2);
```

```
}
```

su1



su2





su1 = su2 = 5;

- |   |  |                   |
|---|--|-------------------|
| ① | printf("su1 + 1 = %d\n", su1 += 1);      | //su1 = su1 + 1   |
| ② | printf("su1 - 1 = %d\n", su1 -= 1);      | //su1 = su1 - 1   |
| ③ | printf("su1 * su2 = %d\n", su1 *= su2);  | //su1 = su1 * su2 |
| ④ | printf("su1 / su2 = %d\n", su1 /= su2);  | //su1 = su1 / su2 |
| ⑤ | printf("su1 %% su2 = %d\n", su1 %= su2); | //su1 = su1 % su2 |

	su1	Su2
default	5	5
①	6	5
②	5	5
③	25	5
④	5	5
⑤	0	5

# 논리연산자

## ❖ 참과 거짓을 판별하는 연산

논리 연산자	의 미	사용 예
	논리합(OR)	if(a<=0    a>=65535) ~
&&	논리곱(AND)	if(b>=60 && b<70) ~
!	부정(NOT)	if(!(c%2==1))~

A	B	(OR, +, 합집합)	&&(AND, *, 교집합)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

## 예제

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

```
main()
```

```
{
```

```
    int num;
```

```
    printf( " 점수를 입력하세요... ");
```

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

```
    if ( !((num > 0) && (num <= 100))){
```

```
        printf("입력 범위를 벗어 났습니다.\n");
```

```
    }
```

```
}
```

# 증감연산자

❖ 피연산자를 1씩 증가 혹은 감소하는 기능

증감연산자	의 미	사용 예
++	피연산자의 값을 1만큼 증가	++a 또는 a++
--	피연산자의 값을 1만큼 감소	--a 또는 a--

❖ 전치와 후치에 따른 연산자 비교

- 전치 : ++a로 표기하며  $a=a+1$ 을 먼저 처리한다
- 후치 : a++로 표기하며 a의 데이터를 사용한 후  $a=a+1$ 을 처리한다.

## 예제

```
#include <stdio.h>
main()
{
    int su1;
    float su2;

    su1=5; ++su1;
    printf("++su1 = %d\n", su1);
    su1=5; su1++;
    printf("su1++ = %d\n", su1);
    su2=12.3; ++su2;
    printf("++su2 = %f\n", su2);
    su2=12.3; su2++;
    printf("su2++ = %f\n", su2);
}
```

## 예제

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

```
main()
```

```
{
```

```
    int su1, su2, su3;
```

```
    su1 = 10;
```

```
    su2 = ++su1;
```

```
    su1 = 10;
```

```
    su3 = su1++;
```

```
    printf("su2 = %d\n", su2);
```

```
    printf("su3 = %d\n", su3);
```

```
}
```

## 예제

```
#include <stdio.h>
int main()
{
    int a=5, b=6, c=10, d;
    d=++a * b--;
    printf("a = %d, b = %d, d = %d\n", a, b, d);
    d=a++ + ++c - b--;
    printf("a = %d, b = %d, c = %d, d = %d\n", a, b, c, d);
    a=1;
    b=0;
    d=a++ || ++b * d-- / ++c;
    printf("a = %d, b = %d, c = %d, d = %d\n", a, b, c, d);
    d=b++ && ++a / ++c * d++;
    printf("a = %d, b = %d, c = %d, d = %d\n", a, b, c, d);
}
```

# 조건연산자

- ❖ 조건식에 따라 참인 경우 앞부분의 식을 거짓인 경우 뒷부분을 수행
- ❖ 사용 예
  - (조건식) ? 참인 경우 : 거짓인 경우 ;
  - 조건식이 참인 경우 ?뒤의 내용을 거짓일 경우 : 뒤의 내용을 실행

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

```
main()
{
    int su;
    su = 8;
    (su % 2 == 0) ? printf("%d : 짝수\n", su): printf("%d : 홀수\n", su);
    su = 9;
    (su % 2 == 0) ? printf("%d : 짝수\n", su): printf("%d : 홀수\n", su);
}
```



# 비트연산자

❖ 10진수를 2진수로 변환하여 각 비트별로 논리/이동 연산을 한다.

비트연산자	의미
	비트 단위 논리합(OR)
&	비트 단위 논리곱(AND)
^	비트 단위 배타적 논리합(XOR)
~	비트 부정(NOT)
<<	비트 좌측 이동(Left Shift)
>>	비트 우측 이동(Right Shift)

## 예제

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

```
main()
```

```
{
```

```
    int su1 = 12, su2 = 7, result;
```

```
    result = su1 | su2;
```

```
    printf("su1과 su2의 논리합 : %d\n", result);
```

```
}
```

	1	1	0	0
	0	1	1	1
<hr/>				
	1	1	1	1

## 예제

```
#include <stdio.h>
main()
{
    int su1 = 15;
    su1 = ~su1;
    printf("su1의 비트 부정 : %d\n", su1);
}
```

~      0000 0000   0000 0000   0000 0000   0000 1111  
         1111 1111   1111 1111   1111 1111   1111 0000

-16

## 예제

```
#include <stdio.h>
main()
{
    int op = 30, result;
    result = op << 3;
    printf("30을 좌측으로 3비트 이동시킨 결과 = %d\n", result);
}
```

<< 3	0000 0000	0000 0000	0000 0000	0001 1110
	0000 0000	0000 0000	0000 0000	1111 0000

# 기타연산자

- ❖ sizeof 연산자 : 크기를 바이트 단위로 표기
- ❖ 콤마연산자
  - 식들을 콤마로 구분하여 좌측부터 실행하며 가장 우측식이 연산결과가 된다.

- ❖ 예제

```
#include <stdio.h>
main()
{
    int a=0, b=0, c=0;
    printf("정수의 크기는 %d입니다. \n", sizeof(int));
    a=( b=3, c=5, b+c);
    printf("a = %d, b = %d, c = %d\n", a, b, c);
}
```

## 예제

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

```
main()
```

```
{
```

```
    int su1 = 123;
```

```
    char ch = 'a';
```

```
    float su2 = 12.345;
```

```
    printf("변수 su1의 크기      : %d byte\n", sizeof(su1));
```

```
    printf("문자형 자료형의 크기   : %d byte\n", sizeof(char));
```

```
    printf("수식(su1+su2)의 크기    : %d byte\n", sizeof(su1+su2));
```

```
    printf("실수 1.23456 의 크기    : %d byte\n", sizeof(1.23456));
```

```
    printf("문자형 상수 'a'의 크기 : %d byte\n", sizeof('a'));
```

```
}
```

## 예제

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

```
main()
```

```
{
```

```
    int su1, su2, su3;
```

```
    float result;
```

```
    result = (su1 = 5, su2 = su1+2, su2++, su3 = su2/su1);
```

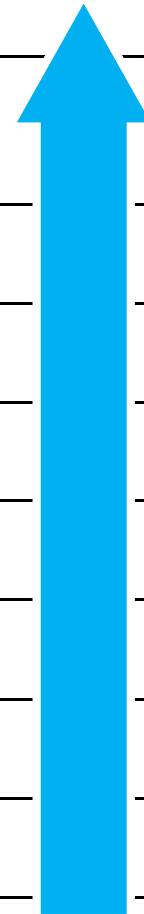
```
    printf(" su1 = %d\n su2 = %d\n"
```

```
           " su3 = %d\n result = %f\n", su1, su2, su3, result);
```

```
}
```

# 연산자 우선순위

연산자	연산순서	우선순위	비고
( ), [ ], ->, .(점)	좌에서 우		
sizeof, (type), &, *, -(단항), +(단항), --, ++, ~, !	좌에서 우		단항
*(곱셈), / , %, +, /	좌에서 우		산술
<<, >>	좌에서 우		비트
<, <=, >, >=, ==, !=	좌에서 우		비교
&, ^,	좌에서 우		비트
&&,	좌에서 우		논리
? :	우에서 좌		삼항
%=, /=, *=, -=, +=, =	좌에서 우		대입
,	좌에서 우		coma





## 문제

- ❖ 수를 입력 받아 짝,홀수를 구분하여 출력하시오
- ❖ 수를 입력 받아 3의 배수이면 출력하시오
- ❖ 두수를 입력 받아 큰 수를 출력하시오
- ❖ 세수를 입력 받아 큰 수를 출력하시오
- ❖ 두수를 입력 받아 큰 수가 짝수이면 출력하시오
- ❖ 두수를 입력 받아 합이 짝수이고 3의 배수인 수를 출력하시오