

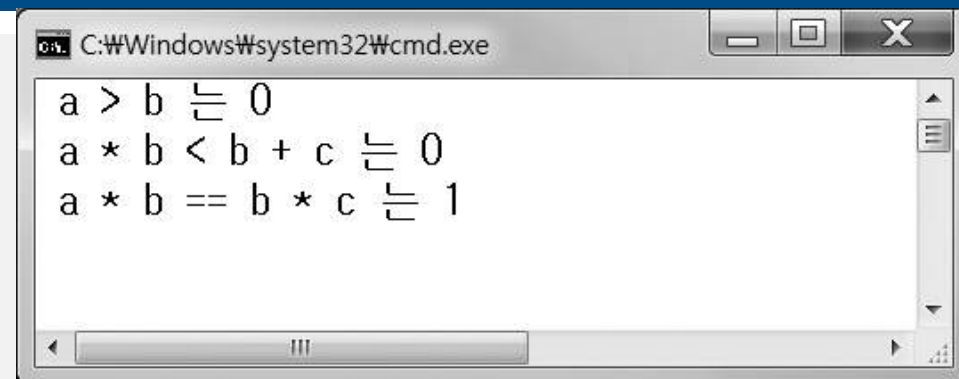
## ● 관계 연산자

- 숫자 2개를 비교하는 연산자로 연산의 결과는 항상 참(true) 또는 거짓(false)

구분	연산자	의미	수학적 표현	C++ 표현	C++ 결과
항등 연산자	==	좌측이 우측과 같다.	=	2 == 4	false
	!=	좌측이 우측과 같지 않다.	≠	2 != 4	true
비교 연산자	>	좌측이 우측보다 크다.	>	3 > 2	true
	>=	좌측이 우측보다 크거나 같다.	≥	2 >= 3	false
	<	좌측이 우측보다 작다.	<	4 < 5	true
	<=	좌측이 우측보다 작거나 같다.	≤	4 <= 4	true

# (예제) 관계 연산자 사용하기

```
01 #include <iostream>
02 using namespace std;
03 int main()
04 {
05     int a = 3, b = 5, c = 3;
06     bool istrue;
07     istrue = a > b;
08     cout << " a > b 는 " << istrue << "\n";
09
10     istrue = a * b < b + c ;
11     cout << " a * b < b + c 는 " << istrue << "\n";
12
13     istrue = a * b == b * c ;
14     cout << " a * b == b * c 는 " << istrue << "\n";
15
16     return 0;
15 }
```



C:\Windows\system32\cmd.exe

```
a > b 는 0
a * b < b + c 는 0
a * b == b * c 는 1
```

## ● 논리 연산자

- 일련의 조건을 모두 만족하는 경우나 일부만 만족하는 경우를 구별하는 데 사용

종류	설명
&&	논리곱 연산자(logical AND operator)
	논리합 연산자(logical OR operator)
!	단항 논리부정 연산자(logical NOT operator)

[논리 연산자 &&와 ||의 진리표]

값		&&	
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

[논리 연산자 !의 진리표]

값	!
0	1
1	0

```

01 include <iostream>
02 using namespace std;
03 int main()
04 {
05     int num=85;
06     bool istrue;
07
08     istrue = ! num == 70;
09     cout<< " ! num == 70 = " << istrue << "\n";
10
11     istrue = ! (num == 70);
12     cout<< " ! (num == 70) = " << istrue << "\n";
13
14     istrue = num >= 80 && num < 90;
15     cout<< " num >= 80 && num < 90 = " << istrue << "\n";
16
17     num=60;
18     istrue = num >= 80 && num < 90;
19     cout<< " num >= 80 && num < 90 = " << istrue << "\n";
20
21     istrue = num >= 90 || num < 80;
22     cout<< " num >= 90 || num < 80 = " << istrue << "\n";
23
24     num=85;
25     istrue = num >= 90 || num < 80;
26     cout<< " num >= 90 || num < 80 = " << istrue << "\n";
27
28     return 0;
29 }

```

The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is as follows:

```

! num == 70 = 0
! <num == 70> = 1
num >= 80 && num < 90 = 1
num >= 80 && num < 90 = 0
num >= 90 || num < 80 = 1
num >= 90 || num < 80 = 0

```

## ● 증감 연산자

- 변수의 값을 1만큼 증가시키나 1만큼 감소

연산자	형태	명칭	의미
++	++a	전위 증가 연산자	연산 전에 a값 증가
	a++	후위 증가 연산자	연산 후에 a값 증가
--	--a	전위 감소 연산자	연산 전에 a값 감소
	a--	후위 감소 연산자	연산 후에 a값 증가

## ● 대입 연산자

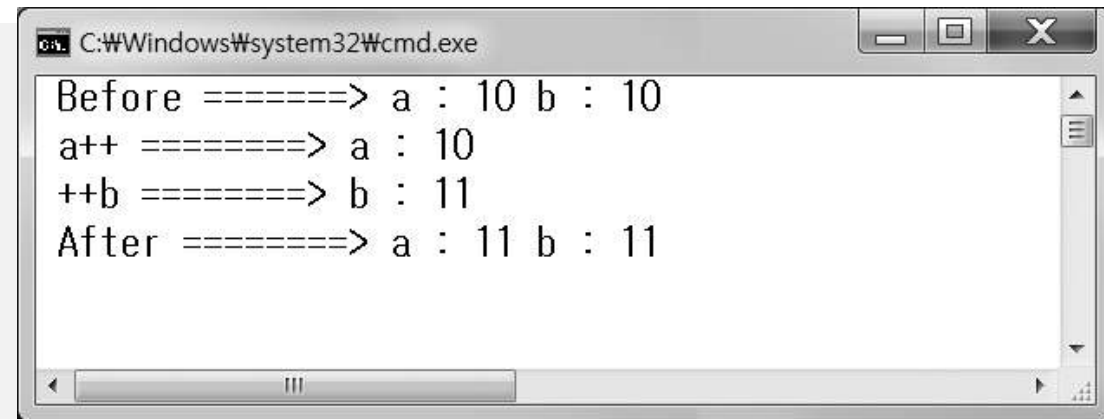
- 대입 연산자 왼쪽에 기술한 변수에 대입 연산자 오른쪽에 기술한 식의 결과를 저장

연산자	사용 예	같은 의미의 수식
+=	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

연산자	사용 예	같은 의미의 수식
&=	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

# (예제) 증가 연산자 사용하기

```
01 #include <iostream>
02 using namespace std;
03 int main()
04 {
05     int a=10, b=10;
06     cout<< " Before =====> a : " << a << " b : " << b << "\n";
07     cout<< " a++ =====> a : " << a++ << "\n";
08     cout<< " ++b =====> b : " << ++b << "\n";
09     cout<< " After =====> a : " << a << " b : " << b << "\n";
10
11     return 0;
12 }
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
Before =====> a : 10 b : 10
a++ =====> a : 10
++b =====> b : 11
After =====> a : 11 b : 11
```

## ● Quiz

```
int x = 0;  
int nextX = 0;
```

### 실행결과

```
nextX = ++x;  
cout << "nextX=" << next << " x=" << x;
```

nextX=1, x=1

```
nextX = x++;  
cout << "nextX=" << next << " x=" << x;
```

nextX=0, x=1

```
nextX = --x;  
cout << "nextX=" << next << " x=" << x;
```

nextX=-1, x=-1

```
nextX = x--;  
cout << "nextX=" << next << " x=" << x;
```

nextX=0, x=-1

## ● Quiz

```
int data = 5;  
int result = 0;
```

실행결과

```
result = data%2;  
cout << "result=" << result;
```

result=1

```
result = data<5;  
cout << "result=" << result;
```

result=0

```
result = data<4+3;  
cout << "result=" << result;
```

result=1

```
result = data++ + 2;  
cout << "result=" << result << " data=" << data;
```

result=7, data=6

```
result = data==5;  
cout << "result=" << result << " data=" << data;
```

result=1, data=5