



06

강

식과 연산자

C 언어

학습목표

▶ C 언어의 식과 연산자에 대해 이해할 수 있다.



EBS tip

1 식

(1) 식의 구성

- ① 식은 연산자와 피연산자로 이루어진다.
- ② 연산자는 미리 정의된 기호를 사용한다.
- ③ 피연산자에는 상수, 변수, 함수 호출 등이 올 수 있다.
- ④ 식의 종류에는 산술식, 관계식, 논리식 등이 있다.

(2) 식의 예

수식	결과값
$-3 + 5$	2
$2 * 6$	12
$7 > 4$	1
$5 + (a = 2 + 4)$	11
$(3 < 4) \&\& (1 < 3)$	1

2 연산자

(1) 대입 연산자

- ① '='을 기준으로 우변의 값 또는 식의 결과를 좌변의 변수에 기억한다.
- ② 형식

변수 = 값 또는 식

■ 대입 연산자를 사용할 때 좌변은 반드시 변수이어야 하고, 우변은 상수, 변수, 식, 함수 호출 등이 올 수 있다.

예 대입 연산자를 이용한 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int a, b, c; a=5; /* 변수=상수(초기화를 의미) */ b=a; /* 변수=변수(치환을 의미) */ c=8-a; /* 변수=수식(연산을 의미) */ printf("%d %d %d", a, b, c); }</pre>	5 5 3

[프로그램 해설]

변수 a에 5를 기억한 후 그 값을 변수 b에 다시 기억한다. 따라서 a와 b의 값은 같으며, 8에서 변수 a의 값 5를 뺀 결과를 변수 c에 기억한다. 마지막으로 printf() 함수에 의해 변수 a, b, c의 값을 출력한다.

(2) 산술 연산자

- ① 사칙 연산과 나머지 연산자가 있다.
- ② 종류

+	덧셈 연산자	-	뺄셈 연산자
*	곱셈 연산자	/	나눗셈 연산자
%	나머지 연산자		

- ③ 정수 간의 나눗셈 결과는 정수값이 되므로 소수 아래 값은 버려진다. 따라서 정확한 값을 얻으려면 적어도 하나 이상의 변수를 실수로 선언해야 한다.

예	프로그램 소스 코드	실행 결과
	<pre>#include <stdio.h> void main() { int a, b, c; a=8-(2*3); b=7/a; c=7%a; printf("%d %d %d", a, b, c); }</pre>	2 3 1

(3) 관계 연산자

- ① 값의 크고 작음을 비교하는 연산자이다.
- ② 종류

>	크다.	<=	작거나 같다.
>=	크거나 같다.	==	같다.
<	작다.	!=	같지 않다.

예 관계 연산자를 사용한 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int a=7, b=5, c=-3; if(a>b) printf("a가 크다. \n"); if(b==c) printf("b와 c는 같다."); if(b!=c) printf("b와 c는 다르다."); }</pre>	a가 크다. b와 c는 다르다.

(4) 논리 연산자

- ① 논리 연산에서는 입력되는 논리 값이 0이면 거짓으로, 0이 아닌 값은 참으로 인식하며, 연산 결과의 출력값은 참이면 1로 거짓이면 0으로 기억된다.
- ② 논리 연산자에는 &&, ||, ! 가 있다.



- 정수끼리의 나눗셈은 몫만 취하고 소수점 이하의 값은 버린다.

- 관계 연산자는 논리식과 비교 문에서 많이 사용한다.

- 대부분의 프로그래밍 언어에서 같음을 나타낼 때 '='를 사용하지만, C 언어에서는 '=='을 사용한다.

- if(a) b는
a가 참이면 b를 실행하라는 뜻이다.

- '&&'은 논리곱, '||'은 논리합, '!'은 논리 부정을 의미한다.



A	B	F
거짓	거짓	거짓
거짓	참	거짓
참	거짓	거짓
참	참	참

▲ (A && B)의 결과

A	B	F
거짓	거짓	거짓
거짓	참	참
참	거짓	참
참	참	참

▲ (A || B)의 결과

A	F
거짓	참
참	거짓

▲ (!A)의 결과

(5) 증가·감소 연산자

① 증가 연산자(++): 변수에 1을 더할 때 사용한다.

- 전위형 증가 연산자: 증가시킬 변수 앞에 ++를 표시한 형태로, 먼저 값을 1 증가시킨 후에 다음 연산을 한다.

예) `n=5; a=++n;` → 결과는 `a=6, n=6`이다.

- 후위형 증가 연산자: 증가시킬 변수 뒤에 ++를 표시한 형태로, 앞의 연산을 먼저 한 후에 값을 증가시킨다.

예) `n=5; a=n++;` → 결과는 `a=5, n=6`이다.

② 감소 연산자(--): 변수의 값에서 1을 뺄 때 사용한다.

- 전위형 감소 연산자: 감소시킬 변수 앞에 --를 표시한 형태로, 먼저 값을 1 감소시킨 후에 다음 연산을 수행한다.

예) `n=5; a=--n;` → 결과는 `a=4, n=4`이다.

- 후위형 감소 연산자: 감소시킬 변수 뒤에 --를 표시한 형태로, 연산을 먼저 한 후에 값을 1 감소시킨다.

예) `n=5; a=n--;` → 결과는 `a=5, n=4`이다.

■ `n`을 먼저 1 증가시킨 후 `a`에 대입한다.

■ `a`에 대입을 먼저 한 후 `n`을 1 증가시킨다.

■ `n`을 먼저 1 감소시킨 후 `a`에 대입한다.

■ `a`에 대입을 먼저 한 후 `n`을 1 감소시킨다.

(6) 복합 배정 연산자

① 복합 배정 연산자는 산술 연산을 한 후에 대입 연산을 해야 하는 표현 식을 하나의 연산자로 간결하게 표시한 것을 말한다.

② 종류

연산자	사용 예	의미
<code>+=</code>	<code>a+=b</code>	<code>a=a+b</code>
<code>-=</code>	<code>a-=b</code>	<code>a=a-b</code>
<code>*=</code>	<code>a*=b</code>	<code>a=a*b</code>
<code>/=</code>	<code>a/=b</code>	<code>a=a/b</code>
<code>%=</code>	<code>a%=b</code>	<code>a=a%b</code>

예 복합 배정 연산자를 사용한 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int a=7, b=5, c=3, d=2; a+=d; printf("%d \n", a); b*=d; printf("%d \n", b); c%=d; printf("%d", c); }</pre>	<pre>9 10 1</pre>

(7) 조건 연산자

- ① 조건 연산자는 if~else 구문을 간결하게 표현하기 위하여 사용한다.
- ② 형식

수식1 ? 수식2 : 수식3

- 수식1은 통상적으로 논리식이어야 한다.
- 수식1의 결과가 참이면 수식2를, 그렇지 않으면 수식3을 실행한다.

예 조건 연산자를 사용한 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio,h> void main() { int a=7, b=5, max; max=(a>b)?a:b; printf("%d", max); }</pre>	7

[프로그램 해설]

(a>b) → (7>5) 이므로 참이 되어 max에 a값을 대입한 후에 max값을 출력한다.

(8) 형 변환

- ① 서로 다른 자료형을 갖는 변수나 상수를 혼용하여 문장과 수식을 구성할 때 형 변환이 일어난다.
- ② 형 변환 기본 규칙
 - 연산되는 두 개의 피연산자 중 높은 순위의 자료형으로 변환된다.
 - 자료형의 우선순위는 기억 장소의 크기로 결정된다.
(int < long < float < double)
 - 대입문에서 연산의 결과는 값을 저장하는 변수의 형으로 변환된다.
- ③ 캐스트(Cast) 연산자
 - 프로그래머가 형 변환을 명시적으로 지정할 때 사용한다.
 - 형식

(자료형)변수 / 상수

예 정수형을 실수형으로 변환하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio,h> void main() { int a=5, b; float c, d; b=a/2; c=a/2; d=(float)a/2; printf("%d, %f, %f", b, c, d); }</pre>	2, 2.000000, 2.500000



3항 연산자라고도 한다.

■ 정수끼리의 나눗셈 연산은 정수인 몫을 취하지만, 정수와 실수의 나눗셈 연산은 소수점 이하의 값까지 모두 얻을 수 있다.

● 프로그램 해설

- ㉠ $c=a/2$ 는 a가 정수이고 2가 정수이므로 연산 결과는 몫인 2를 얻는다.
- ㉡ 따라서 원하는 결과(소수점 이하의 값)를 얻기 위해서는 $c=a/2.0$ 과 같이 소수점을 표현하거나, (float)a와 같이 정수형인 a를 실수형으로 바꾸어 주는 캐스트 연산자를 사용해야 한다.



(9) 연산자 우선순위

C 언어에서 사용하는 연산자들의 우선순위는 다음과 같다.

우선순위	연산자	기능(설명)
1. 최상위 연산자	[] ()	배열 첨자 함수 호출 연산자
2. 단항 연산자	! + / - ++ -- &	논리 부정(NOT) 양수/음수 증가 연산자 감소 연산자 주소
3. 산술 연산자	* / %	곱셈 나눗셈 나머지
4. 산술 연산자	+ -	덧셈 뺄셈
5. 관계 연산자	< <= > >= == !=	작다. 작거나 같다. 크다. 크거나 같다. 같다. 같지 않다.
6. 논리 연산자	&& 	논리곱 논리합
7. 3항 연산자	?:	조건 연산자
8. 복합 배정 연산자	+=, -=, *=, /=, %=	

예 우선순위에 따른 연산 순서

c = a > 8 - (a * 2) % 5;
↑ ↑ ↑ ↑ ↑
① ② ③ ④ ⑤

연산순위는 ④ → ⑤ → ③ → ② → ①이다.



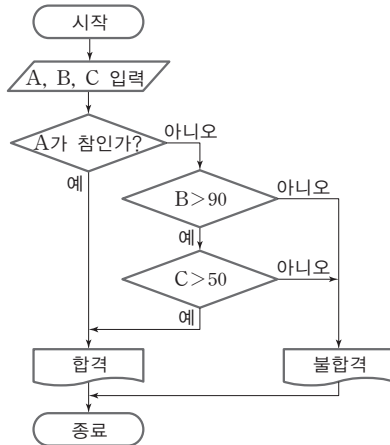
기출 모의고사

정답 및 해설 p. 6

1

2010년 6월 시행 평가원 모의평가

다음 순서도에 의해 입사 여부가 결정된다. “합격”을 출력하기 위한 조건을 논리식으로 표현할 때 옳은 것은?



A : 특별 채용 대상(참 또는 거짓)
B : 입사 시험 점수(0~100)
C : 면접 점수(0~100)

- ① $A \parallel (B > 90 \ \&\& \ C > 50)$
- ② $A \parallel (B > 90 \parallel C > 50)$
- ③ $A \ \&\& \ (B > 90 \ \&\& \ C > 50)$
- ④ $!A \parallel (B > 90 \ \&\& \ C > 50)$
- ⑤ $!A \ \&\& \ (B > 90 \parallel C > 50)$

2

2010학년도 대수능

다음 [조건]에 따라 n 개의 사탕을 묶음과 잔여분으로 표현할 때 a , b 를 구하는 연산식으로 옳은 것은?

[조건]

- 정수형 변수 n 의 입력 값은 자연수이다.
- 한 묶음은 10개이고, 잔여분은 10개 미만이다.
- 묶음의 수를 계산하여 정수형 변수 a 에 저장한다.
- 잔여분을 계산하여 정수형 변수 b 에 저장한다.



- ① $a = n/10; b = n\%10;$
- ② $a = n/10; b = n+10;$
- ③ $a = n+10; b = n\%10;$
- ④ $a = n*10; b = n\%10;$
- ⑤ $a = n\%10; b = n*10;$

3

다음은 신학기 학급 편성 프로그램 알고리즘 내용 중 일부이다. 밑줄 친 부분을 만족시킬 수 있는 조건식으로 옳은 것은?

- 단계 1 : $number = 0, class = 1, student = 0$
- 단계 2 : $number = number + 1, student = student + 1$
- 단계 3 : $student > 265$ 이면 종료
- 단계 4 : $number$ 가 35이면 $number = 0,$
 $class = class + 1$
- 단계 5 : 단계 2로 이동

- ① $number = number + 35;$
- ② $number = number - 35;$
- ③ $number = number * 35;$
- ④ $number = number / 35;$
- ⑤ $number = number \% 35;$

4

다음 프로그램의 출력 결과는?

```

#include <stdio.h>
void main() {
    int a, b, c, s;

    a=7;
    b=a/2;
    c=a%2;
    s=b-c;
    printf("%d", s);
}
  
```

- ① 0
- ② 1
- ③ 2
- ④ 3
- ⑤ 4

5 다음의 [알고리즘]을 진행한 후의 출력 결과로 옳은 것은?

[알고리즘]

단계 1 : 변수 a에 1을 대입한다.
 단계 2 : a를 2로 나눈 나머지가 0이 아니면 단계 5로 이동한다.
 단계 3 : a를 3으로 나눈 나머지가 0이 아니면 단계 5로 이동한다.
 단계 4 : a를 출력한다,
 단계 5 : a=a+1
 단계 6 : a가 10보다 작으면 단계 2로 이동하고 그렇지 않으면 종료한다.

- ① 2 4 6 8 ② 3 6 9 ③ 2 3 4 6 8 9
 ④ 6 ⑤ 6 9

6 다음 프로그램을 바르게 분석한 학생을 <보기>에서 모두 고른 것은?

```
#include <stdio.h>
void main() {
    int a=8, b=3, k, s=0;

    scanf("%d", &k);
    if (k==0) s=a+b;
    if (k==1) s=a-b;
    if (k==2) s=a*b;
    if (k==3) s=a/b;
    printf("%d", s);
}
```

보기

영희 : 입력값 k가 0~3의 범위일 경우 k의 값에 따라 사칙연산을 하는 프로그램이야.
 길동 : k가 0보다 작거나 3보다 크면 문법적 오류가 발생하게 돼.
 철수 : k에 2를 입력하면 24가 출력되겠군.

- ① 영희 ② 길동 ③ 영희, 철수
 ④ 길동, 철수 ⑤ 영희, 길동, 철수

7 다음 프로그램은 두 변수 a, b의 값을 서로 바꾸어 저장하는 프로그램이다. (가)의 내용으로 옳은 것은?

프로그램 소스 코드	<pre>#include <stdio.h> void main() { int a=5, b=3, temp; (가) printf("a=%d, b=%d", a, b); }</pre>
	실행 결과 a=3, b=5

- ① a=b;
 b=a;
 ② a=b;
 temp=a;
 b=temp;
 ③ temp=a;
 a=b;
 b=temp;
 ④ temp=b;
 a=b;
 b=temp;
 ⑤ a=b;
 temp=b;
 b=a;