C Programming Assignment / Week 4

정리노트 #3

연산자의 종류

“연산자”는 어떠한 이미 정의된 연산을 수행하는 기호로, 하나의 연산식을 만드는 데 사용한다.

그 연산에 참여하는 변수나 값은 “피연산자”라고 한다.

- 대입 연산자

대입 연산자는 = 기호를 사용한다. 수학과는 달리, 프로그래밍에서는 대입 연산자를 사용할 때에는 연산의 순서가 있다. 대입 연산자를 기준으로 오른쪽에 있는 수식을 먼저 실행 후, 그 결과를 왼쪽 값에 대입한다.

ex) x = 5 + 6 입력 시 x 에 11 이라는 값이 대입됨.

- 산술 연산자

산술 연산자는 +,-,*,/ 와 같이 산술 연산을 수행하는 연산자다. 연산자가 다루는 피연산자의 개수에 따라 단항, 이항 연산자로 구분된다.

- 증감 연산자

증감 연산자는 ++와 --가 있으며, 피연산자의 앞에 붙으면 전치 증가, 뒤에 붙으면 후치 증가라고 한다. 연산의 처리 속도를 빠르게 하기 위하여 사용한다.

- 이항 연산자

이항 연산자는 피연산자가 2 개 필요한 산술 연산자로, 사칙연산과 나머지 연산자를 포함한다.\

+,-,*,/,% 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지를 구하는 연산자가 있다.

- 관계 연산자

관계 연산자는 두 수 사이의 대소 관계 및 특정 조건을 검사할 때 사용하는 연산자이다. 참이면 true or 1, 거짓이면 false or 0 으로 표시한다.

>, <, >=, <= 는 수학에서의 의미와 같고, 같다의 의미의 등호는 ==, 같지 않다는 의미는 !=로 표시한다.

- 논리 연산자

논리 연산자는 조건 여러 개를 결합하여 판정하는 연산자이다. AND, OR, NOT 의 논리 연산을 수행한다. 참은 1, 거짓은 0 으로 결과값이 표시된다.

&& = and 의 의미로, 양 쪽 피연산자가 모두 참일 경우에는 1, 거짓일 경우에는 0

|| = or 의 의미로, 양 쪽 피연산자 중 하나라도 참일 경우에는 1, 둘다 아닐 경우에는 0

! = !뒤의 숫자가 참이면 1, 거짓이면 0

- 비트 연산자

피연산자의 정숫값을 비트 단위로 두고 논리 연산을 수행하는 연산자이다. AND, OR, NOT 의 기본 논리식으로 이루어진다. n 비트로는 2n 개의 데이터를 나타낼 수 있다.

비트 연산에는 shift 연산이 존재하는데, 비트를 오른쪽이나 왼쪽으로 이동시켜 비트값을 감소시키거나 증가시킬 수 있다.

- 축약 연산자

축약 연산자는 연산 2 개를 동시에 수행하여 값을 할당하는 연산자이다.

연산자의 종류로는 +=, -=, *=, /= 등등이 있으며 프로그램을 좀 더 간결하게 만들기 위해 사용한다.

ex) a += b // a 와 b 를 더하여 a 에 대입한다.

- 콤마 연산자

콤마 연산자는 수식 2 개를 문장 1 개로 표현할 때 사용한다. 주로 변수 선언에 많이 쓰인다.

연산자 중 우선순위는 가장 낮으며 왼쪽부터 오른쪽으로 연산을 수행한다.

ex) x = 1 + 2, 3 – 4

여기서 들은 의문이 있다.

위의 예시에서 x = 1 + 2, 3 – 4 의 수식에서는 printf 를 했을 때 앞의 값을 출력하는데

저 x 의 수식 2 개를 괄호로 묶으면 Ex) (1 + 2, 3 – 4) 뒤의 함숫값이 출력된다는 점이었는데.

가정으로, 위의 방식에선 1 + 2 의 수식만을 인식 해, 뒤에 뭘 쓰든 말든 인식을 할 수 없다라는 가정과, 괄호로 묶은 두 수식은 그 두 수식이 하나의 항이 되어 뒷수식의 값을 출력한다는 가정인데.

그럼 괄호로 묶었을 때, 앞에 있는 1 + 2 의 값은 왜 사라지는 것인가? 에 대한 의문이 사라지지 않았다.

- sizeof 연산자

sizeof 연산자는 변수나 자료형의 크기를 알고 싶을 때 사용한다. 크기 단위는 바이트이다.

ex) sizeof(int); // 결과 값은 4

- 연산자 우선순위

연산자에는 우선순위가 높은 것부터 연산하는 특성이 있는데, 이 우선순위는

- 단항 > 산술 > 이동 > 관계 > 비트 > 논리 > 조건 > 대입 의 순서다.

연습문제 3

```
#include <stdio.h>
int main(void)
{
        3int a,b,c;
        a = 10; b=-20; c=30;

        a = ++b; //If, printf : a = 21 ,b = 21

        c= b++; //If printf c = 21, b = 22
        printf("a= %d, b= %d, c= %d₩n₩n", a,b,c);

        a  = ++b + ++c //a = 45 b= 23, c=22
        printf("a = ++b + ++c 문장 실행 후₩n");
        printf("a = %d. b= %d, c = %d₩n₩n", a,b,c);
```

```
	a = b++ + c++; //a = 45, b = 24, c = 23
	printf("a = b++ + c++ 문장 실행 후₩n");
	printf("a = %d, b = %d, c = %d₩n₩n", a,b,c);

	return 0;
}
```

이 문제에서는 증감 연산자인 ++(변수), (변수)++의 출력 방식을 잘 알아야 하는 문제이다.

첫번째 연산: a = ++b 의 경우 ++(변수)는 변숫값을 1 만큼 먼저 증가시킨 후, 최종 변숫값을 수식에 적용하는 방식이므로 a 와 b 는 같은 값인 21 을 공유한다.

두번째 연산: c = b++ 에서 (변수)++의 경우는 변숫값을 수식에 먼저 적용을 한 후, 최종 변숫값을 1 만큼 증가시키는 의미로 c 는 b 연산 전 값인 21 을 가지고 b 는 1 을 더한 값을 가지므로 22 를 가진다.

세번째 연산: a = ++b + ++c 에서는 증감 연산자와 산술 연산자가 섞인 연산인데 이도 똑같이 ++(변수)의 값을 먼저 실행을 하고 산술 연산자를 실행한다. 즉 b 와 c 에 1 씩을 더하고 그 둘을 더한 값인 45 를 a 에 저장하고 b 는 23, c 는 22 를 가진다.

네번째 연산: a = b++ + c++ 이것도 세번째 연산에서 증감 연산자의 서순만 바뀐 상황이다. 세번째 연산과 마찬가지로 현재 b 와 c 의 값을 더한 값인 45 는 a 에 저장되고 b 와 c 는 연산을 하고 난 후, 최종 변숫값을 1 더하므로 b 는 23, c 는 22 를 가진다.

다음으론 127 쪽 예제 4 번을 풀어보려 한다.

정수 3 개를 입력받아 합을 계산한 뒤 합이 100 보다 크면 '합격'을, 그렇지 않으면 '불합격'을 출력하는 프로그램을 작성하시오.

이 문제같은 경우에는 입력받는 정수 3 개는 scanf 를, 그리고 관계 연산자를 이용하여 참거짓을 판별하여 if else 로 합격 또는 불합격을 출력하게 만들면 된다.

연습문제 4

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

int main(void)
{
int score1, score2, score3, grade; //입력받을 정수 3 개 score 1, 2, 3 과 그 합산 grade 추가

printf("첫 번째 점수를 입력하세요: ₩n");
scanf("%d", &score1); // 첫 번째 점수 입력받기
printf("두 번째 점수를 입력하세요: ₩n");
scanf("%d", &score2); // 두 번째 점수 입력받기
printf("세 번째 점수를 입력하세요: ₩n");
scanf("%d", &score3); // 세 번째 점수 입력받기

grade = score1 + score2 + score3 > 100; // 관계 연산자를 이용하여 점수의 총합 grade 가 100 보다 크면 1, 아니면 0 이 나오게 출력
if (grade == 1) // 100 보다 크면(문제의 조건) 1 이 나오므로 합격이 뜨게
printf("합격입니다.");
else // 100 보다 크지 않으면 0 이란 값이 나오고, 이쪽으로 넘어와서 아래의 문구(불합격입니다)가 출력
printf("불합격입니다.");

return 0;
```

}

이렇게 하면 제시된 문제에 부합하는 결과를 얻을 수 있다.