



## C - HW2

임베디드스쿨1기

Lv1과정

2020. 07. 29

김인겸

# 4. C Programing operators

---

1) Arithmetic operater : + - \* / % ++ --

++, --는 변수 앞에 위치하는지 뒤에 위치하는지에 따라 해석이 다르다

ex) int a = 0  
int b = 10;

a=b++; //b값을 먼저 a에 집어넣은 후 b의 값을 +1증가 시키는 연산  
(이 연산이 끝나면 a=10, b=11)

a=++b; // b값을 +1증가 시킨 후 b값을 a에 집어넣는 연산  
(이 연산이 끝나면 a=12, b=12)

# 4. C Programing operators

---

2) Logical operator : && || !

- && : 2개의 변수 중 하나라도 0이 아니면 참.
- || : 2개의 변수 둘 다 0이 아니면 참.
- ! : 문장의 참 거짓을 반전 시키는 용도

3) Relational operator : == != > < >= <=  
문장의 참 거짓을 판단하는 용도.

## 4. C Programing operators

---

3) bitwise operator : & | ^ ~ << >>

& : AND연산

| : OR연산

^ : XOR연산

~ : NOT연산(비트를 반전시키는 용도)

<< : 왼쪽으로 shift하는 연산

>> : 오른쪽으로 shift하는 연산

- Shift연산은  $2^n$ 을 곱하거나 나누는 용도로 사용된다
- 비트연산자를 사용하는 이유는 연산속도를 빠르게하기 위함이다.

4) Assign operator : = += -= \*= /= %= <<= >>= &= ^= |=

-대입연산자를 사용하면 속도가 빨라진다.

# 4. C Programing operators

HW1

$(A+B)\%C$ 는  $((A\%C)+(B\%C))\%C$ 와 같을까?

$(A*B)\%C$ 는  $((A\%C)*(B\%C))\%C$ 와 같을까?

세 수 A,B,C가 주어졌을 때, 위의 네 가지 값을 구하는 프로그램을 작성하시오

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int A, B, C;
6     int value1, value2, value3, value4;
7
8     printf("세 수 A, B, C를 입력하시오\n");
9     scanf("%d %d %d", &A, &B, &C);
10
11     value1 = (A + B) % C;
12     value2 = ((A % C) + (B % C)) % C;
13     value3 = (A * B) % C;
14     value4 = ((A % C) * (B % C)) % C;
15     printf("\n");
16
17     printf("%d\n", value1);
18     printf("%d\n", value2);
19     printf("%d\n", value3);
20     printf("%d\n", value4);
21
22     return 0;
23 }
```

```
세 수 A, B, C를 입력하시오
5 8 4
```

```
1
1
0
0
```

# 5. C programing Decision Making

---

Decision Making이란

특정한 조건에 따라 문장의 실행 순서를 결정하는 과정을 뜻한다.

조건문의 종류

1) if문

if문을 여러번 쓸 경우 문장만 참이라면 if~else문과는 다르게 다양한 조건을 만족시킬 수 있다

2) if ~ else문

여러 조건 중에서 하나의 조건만 만족시킨다. (if는 여러개 가능, if~else는 하나만!)

3) switch ~ case문

If~else문과 마찬가지로 하나의 조건만 충족시키고 싶을 때 사용한다.

switch문은 if~else보다 가독성이 좋고 속도가 일반적으로 빠르다.

# 5. C programing Decision Making

## HW1. 시험점수 출력 프로그램

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     unsigned int grade = 0;
6
7     printf("시험 점수를 입력하시오\n");
8     scanf("%u", &grade);
9
10    if(grade > 100){
11        printf("잘못 입력하셨습니다\n");
12    }else if(grade >= 90){
13        printf("A\n");
14    }else if(grade >= 80){
15        printf("B\n");
16    }else if(grade >= 70){
17        printf("C\n");
18    }else if(grade >= 60){
19        printf("D\n");
20    }else
21        printf("F\n");
22
23    return 0;
24 }
```

```
시험 점수를 입력하시오
100
A
(base) ingyecomkim@ingyecomk
시험 점수를 입력하시오
90
A
(base) ingyecomkim@ingyecomk
시험 점수를 입력하시오
72
C
(base) ingyecomkim@ingyecomk
시험 점수를 입력하시오
50
F
(base) ingyecomkim@ingyecomk
시험 점수를 입력하시오
101
잘못 입력하셨습니다
(base) ingyecomkim@ingyecomk
시험 점수를 입력하시오
-1
잘못 입력하셨습니다
```

# 5. C programing Decision Making

## HW2. 사분면 출력 프로그램

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     float x, y;
6
7     printf("x좌표와 y좌표를 입력하시오\n");
8     scanf("%f %f", &x, &y);
9
10    if(x>0 && y>0){
11        printf("1사분면\n");
12    }else if(x<0 && y>0){
13        printf("2사분면\n");
14    }else if(x<0 && y<0){
15        printf("3사분면\n");
16    }else if(x>0 && y <0){
17        printf("4사분면\n");
18    }
19    return 0;
20 }
```

```
x좌표와 y좌표를 입력하시오
12.5
6
1사분면
(base) ingyeomkim@ingyeomkim
x좌표와 y좌표를 입력하시오
-12.5
6
2사분면
(base) ingyeomkim@ingyeomkim
x좌표와 y좌표를 입력하시오
-12.5
-6
3사분면
(base) ingyeomkim@ingyeomkim
x좌표와 y좌표를 입력하시오
12.5
-6
4사분면
```



# 6. C programing Loop Control

## 반복문의 종류

### 1) while문 :

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a = 10;
6
7     while(a < 20){
8         printf("Value of a is %d", a);
9         a++;
10    }
11
12    return 0;
13 }
```

- 변수 a의 초기값
  - while문이 반복되기 위한 조건
  - 변수a의 변화
- 이 세 가지 요소가 포함되어야 한다

2) do while문 : while문이 조건을 만족시킬 때만 동작하는 반복문이었다면 do while 문은 조건을 검사하기 전에 일단 한 번은 무조건 실행하는 반복문이다. 조건을 먼저 검사하는 while문의 단점을 해소할 수 있다.

## 6. C programming Loop Control

---

- 3) for문 : for문은 변수의 초기값과 조건, 그리고 변수의 변화가 for문 안에 모두 들어가 있는 반복이다  
while문 보다 가독성이 좋다.

## 6. C programing Loop Control

---

반복문을 탈출하거나 생략하는 방법

break문, continue문, goto문이 있으며 이것들은 조건문과 결합되어 사용된다.

- 1) break : 특정 조건이 만족되면 반복문을 빠져나가는 문장.
- 2) continue : 특정 조건이 만족되면 해당 반복문을 한 번 건너뛰는 문장
- 3) goto : 미리 label을 지정해놓고 goto를 쓰면 지정된 label로 이동하는 문장

# 6. C programing Loop Control

## HW1. 구구단 입력 프로그램

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     unsigned int n;
6     int i, value;
7
8     printf("n값을 입력하세요 : ");
9     scanf("%u", &n);
10
11     for(i = 1 ; i <10 ; i++){
12         value = n * i;
13         printf("%u * %d = %d\n", n, i, value);
14     }
15
16     return 0;
17 }
```

```
n값을 입력하세요 : 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
```

# 6. C programing Loop Control

## HW2. 별 찍기

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     unsigned int line ;
6     int i, j;
7
8     printf("몇줄을 출력할 건지 입력하세요 : ");
9     scanf("%u", &line);
10
11     for(i = 0 ; i < line ; i++){
12         for(j = 0 ; j <= i ; j++){
13             printf("*");
14         }
15         printf("\n");
16     }
17
18
19     return 0;
20 }
```

```
(base) ingyeomkim@ingyeomkim-Inspiro
몇줄을 출력할 건지 입력하세요 : 3
*
**
***
(base) ingyeomkim@ingyeomkim-Inspiro
몇줄을 출력할 건지 입력하세요 : 6
*
**
***
****
*****
*****
```

# 6. C programming Loop Control

HW3. 세자리수 x 세자리수

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int num1, num2;
6     int k = 0;
7     int temp ;
8     int value1, value2, value3;
9     int final_value;
10
11     scanf("%d %d", &num1, &num2);
12
13     temp = num2 % 10;
14     k += temp;
15     value1 = num1 * temp;           // num1 * (num2의 일의 자릿수)
16
17     temp = ((num2 - k) % 100) / 10;
18     k += temp;
19     value2 = num1 * temp;          // num1 * (num2의 십의 자릿수)
20
```

밑으로

## 6. C programming Loop Control

```
21  temp = (num2 - k) % 1000 / 100;
22  k += temp;
23  value3 = num1 * temp;           // num1 * (num2의 백의 자릿수)
24
25  final_value = num1 * num2;
26
27  printf("%d\n", value1);
28  printf("%d\n", value2);
29  printf("%d\n", value3);
30  printf("%d\n", final_value);
31
32  return 0;
33 }
34
```

입력	472
	385
출력	2360
	3776
	1416
	181720
	(base) ing
입력	766
	293
출력	2298
	6894
	1532
	224438



# 6. C programming Loop Control

HW3+. 임의의 자연수를 입력받고 출력하는 프로그램

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int num1, num2;
6     int temp;           // 일시적인 값
7     int d = 0;          // 자릿수를 구할 때 사용
8     int i;              // for문 쓸때 사용
9     int t = 1;          // 10의 지수승을 표현할 때 사용
10    int value[9];        // num1 * (num2의 자릿수)를 곱한 값을 넣는 용도
11    int final_value;     // 최종값
12
13    scanf("%d %d", &num1, &num2);
14
15    while(1){            // num2자릿수 구하는 반복문
16        temp = num2 / t; // d값이 곧 num2의 자릿수
17        if(temp == 0)
18            break;
19        d++;
20        t *= 10;
21    }
```

(밑으로)



```

for(i = 1, t = 1 ; i <= d ; i++){ // num1과 (num2의 자릿수)를 곱한 값을
    temp = num2 / t % 10;         // value의 배열에 차례로 넣고 출력하는 반복
    value[i-1] = num1 * temp;
    printf("%d\n", value[i-1]);
    t *= 10;
}
final_value = num1 * num2;
printf("%d\n", final_value);      // 최종값 출력

return 0;
}

```

```

(base) ingyemk
472
385
2360
3776
1416
181720
(base) ingyemk
100
123456
600
500
400
300
200
100
12345600

```