

## <LAB 4>

### 연산자

제출일 : 2022/09/26

이름 : 고려욱

학번 : 201810528

## 1. Lab 4-7)

### 1) 실습 문제

[1] 정수인 천만 이하의 한 수를 입력 받아 우리가 사용하는 단위인 만, 천, 백, 십, 일 단위로 출력하는 프로그램을 작성하시오.

[2] 표준입력으로 두 정수를 입력 받아 큰 수를 작은 수로 나눈 몫과 나머지를 출력하는 프로그램을 작성하시오.

세수를 입력 받아 가장 큰 수를 구하는 프로그램을 조건 연산자를 이용하여 작성하시오.

### 2) 배경 지식

#### [1] 산술연산자

산술연산자는  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ 로 구성되어 있다. 덧셈, 뺄셈, 곱셈은 우리가 실생활에 사용하는 연산자와 동일하게  $+$ ,  $-$ ,  $*$  를 사용한다. 나눗셈은 몫과 나머지를 연산하는 연산자가 분리되어 있다.  $\%$ 는 나머지(modulus)연산자로 피연산자로 정수만 가능하며 실수가 입력되었을 경우 오류가 발생한다.  $/$ 는 몫을 도출하는 연산자로 피연산자로 정수, 실수 모두 사용 가능하다.

#### [2] 조건연산자

조건연산자는 조건에 따라 주어진 피연산자가 결과값이 되는 삼항연산자이다.  $(x ? a : b)$ 로 사용하며  $x$ 에는 조건에 해당하는 연산식이 들어간다. 이후 해당 조건이 참이라면  $a$ 부분에 해당하는 연산이 진행되고 참이 아니라 거짓이라면  $b$ 에 해당하는 연산이 진행된다.

#### [3] 비교연산자

비교연산자는 피연산자들의 크기를 비교하기 위한 연산자로 그 결과값이 참이면 1, 거짓이면 0이 출력되는 연산자이다.  $>$ ,  $<$ ,  $=>$ ,  $=<$ ,  $!=$ ,  $==$  이 사용되며 이항연산자로 두 피연산자 간의 크고 작음 같음과 같지 않음에 대한 정보를 알려준다.

### 3) 소스 코드

[1]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    int num;
    printf("input your num under 10million : ");
    scanf("%d", &num);
    printf("%d만", num / 10000);
    printf("%d천", (num % 10000) / 1000);
    printf("%d백", ((num % 10000) % 1000) / 100);
    printf("%d십", (((num % 10000) % 1000) % 100) / 10);
    printf("%d\n", (((num % 10000) % 1000) % 100) % 10);
    return 0;
}
```

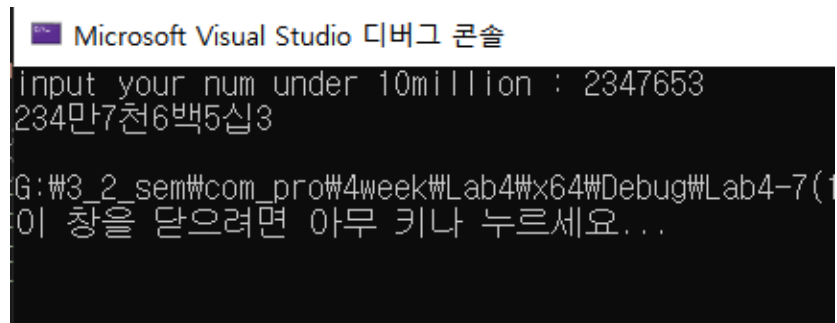
[2]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    int num1, num2;
    printf("input 2 numbers : ");
    scanf("%d %d", &num1, &num2);
    num1 > num2 ? printf("몫 : %d, 나머지 : %d\n", num1 / num2, num1 % num2)
                : printf("몫 : %d, 나머지 : %d\n", num2 / num1, num2 % num1);

    int a, b, c;
    printf("input 3 numbers : ");
    scanf("%d %d %d", &a, &b, &c);
    a > b ? (a > c ? printf("입력한 정수 중에서 가장 큰 수는 %d입니다.\n", a) :
    printf("입력한 정수 중에서 가장 큰 수는 %d입니다.\n", c))
          : (b > c ? printf("입력한 정수 중에서 가장 큰 수는 %d입니다.\n", b) :
    printf("입력한 정수 중에서 가장 큰 수는 %d입니다.\n", c));
    return 0;
}
```

### 4) 실행결과

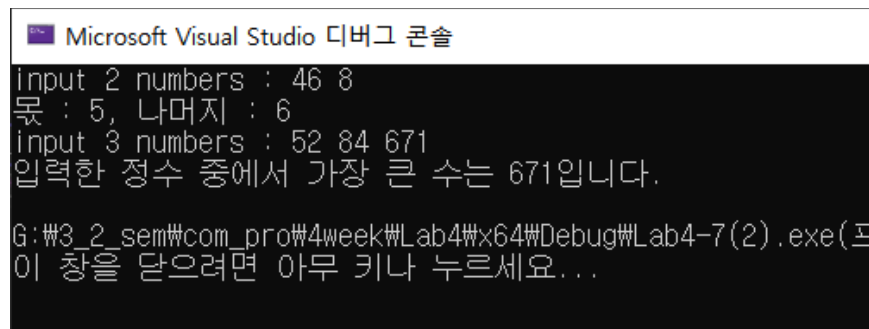
[1]



```
Microsoft Visual Studio 디버그 콘솔
input your num under 10million : 2347653
234만7천6백5십3

G:\#3_2_sem\com_pro\#4week\#Lab4\#x64\Debug\Lab4-7(1)
이 창을 닫으려면 아무 키나 누르세요...
```

[2]



```

Microsoft Visual Studio 디버그 콘솔
input 2 numbers : 46 8
몫 : 5, 나머지 : 6
input 3 numbers : 52 84 671
입력한 정수 중에서 가장 큰 수는 671입니다.
G:\#3_2_sem\com_pro\4week\Lab4\#x64\Debug\Lab4-7(2).exe(프
이 창을 닫으려면 아무 키나 누르세요...

```

## 5) 결과분석

- 나머지와 몫을 나누어서 출력하기 위해서 연산자로 %, / 를 잘 사용하여서 출력하여야 함
- 출력의 결과로 줄바꿈이 들어가면 안되기 때문에 printf를 사용하여 'wn'줄바꿈 문자를 사용하지 않음
- 큰 수와 작은 수에 대한 비교를 위해 조건연산자가 사용되어야 함.
- 세 수에 대한 비교는 처음으로 두 수를 비교하고 참과 거짓에 이중으로 조건연산자를 사용하여 해당 결과로 나온 수와 비교하지 않은 수를 비교하여 출력해야한다.

## 2. Lab 4-8)

### 1) 실습 문제

[1] 조건연산자를 이용하여 임의의 달을 입력 받아 이 달이 상반기이면 "상반기입니다."를 하반기이면 "하반기입니다."를 출력하는 프로그램을 작성하시오.

[2] 다음 조건을 만족하는 프로그램을 작성하시오.

- 원금이 1000000인 경우, 예치 기간을 년 단위로 입력 받아 만기 시 총 금액을 출력
- 만기 시 총 수령액(단리적용) = 원금(1 + 이율(4.5%) \* 년(예치기간))

## 2) 배경 지식

### [1] 조건 연산자

조건연산자는 조건에 따라 주어진 피연산자가 결과값이 되는 삼항연산자 이다. (x ? a : b)로 사용하며 x에는 조건에 해당하는 연산식이 들어간다. 이후 해당 조건이 참이라면 a부분에 해당하는 연산이 진행되고 참이 아니라 거짓이라면 b에 해당하는 연산이 진행된다.

## 3) 소스 코드

### [1]

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

int main(void)
{
    int month;
    printf("input your month : ");
    scanf("%d", &month);
    month > 6 ? printf("하반기입니다.\n") : printf("상반기입니다.\n");

    return 0;
}
```

### [2]

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

int main(void)
{
    int o_money;
    int year;
    printf("input your origin money : ");
    scanf("%d", &o_money);
    o_money >= 1000000 ? printf("input year want to know : ") : printf("money under 1000000");
    o_money >= 1000000 ? scanf("%d", &year) : printf("");
    o_money >= 1000000 ? printf("your money is : %ld\n", o_money / 1000 * (45 * year)) : printf("");
    return 0;
}
```

#### 4) 실행결과

[1]

```

Microsoft Visual Studio 디버그 콘솔
input your month : 8
하반기입니다.
G:\3_2_sem\com_pro\4week\Lab4\64\Debug
이 창을 닫으려면 아무 키나 누르세요...

Microsoft Visual Studio 디버그 콘솔
input your month : 4
상반기입니다.
G:\3_2_sem\com_pro\4week\Lab4\64\Debug
이 창을 닫으려면 아무 키나 누르세요...

```

[2]

```

Microsoft Visual Studio 디버그 콘솔
input your origin money : 1000000
input year want to know : 8
your money is : 1360000
G:\3_2_sem\com_pro\4week\Lab4\64\Debug
이 창을 닫으려면 아무 키나 누르세요...

```

#### 5) 검토의견

- 두번째 문제 계산식 원금 $(1+4.5\%*년)$  실행시에 그대로 코드로 적게 되면 문제가 발생
- 정수와 실수의 곱에서 문제가 생기는 것으로 해석됨. 형변화를 컴파일상에서 자동으로 진행하게 돼서 문제가 발생하는 것으로 해석
- 해결방법으로 식을 풀어서 사용하였는데 더 좋은 방법이 있을 것으로 생각된다.
- 형 변환은 아직 강의에서 다루기 이전이라 배경지식에 서술하지 않음

### 3. Lab 4-9)

#### 1) 실습 문제

[1]

이차원 평면에서 다음 두 점 (3.2, 4.6)와 (-8.3, -2.3)의 중간지점을 출력하는 프로그램

램을 작성하시오.

- $(x_1, y_1)$ 과  $(x_2, y_2)$ 의 중간지점 =  $((x_1+x_2)/2, (y_1+y_2)/2)$

[2]

지불할 금액을 정수로 입력 받아 화폐단위가 각각 몇 개씩 필요한지 출력하는 프로그램을 작성하시오.

- 입력은 최소 천원 단위로 입력
- 화폐단위는 500000, 100000, 5000, 1000 4가지이며, 가능한 큰 화폐단위로 지불

## 2) 배경 지식

[1] 실수의 계산

실수의 나눗셈에는 몫을 알기 위한 나눗셈 연산자 '/'만 사용할 수 있다. 나머지를 알 수 있는 %의 경우 정수만 사용가능한데 그 이유로는 실수를 실수로 나눴을 때 나머지가 정확한 유효한 숫자로 떨어지지 않을 가능성이 높아 오류를 발생하기 때문이다.

## 3) 소스 코드

[1]

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

int main(void)
{
    printf("(%.1f %.1f)Wn", (3.2 + (-8.3)) / 2, (4.6 + (-2.3)) / 2);
    return 0;
}
```

[2]

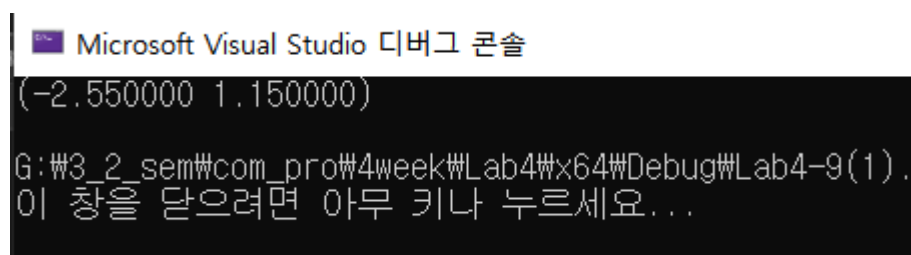
```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
    int money;
    printf("input your money : ");
}
```

```
scanf("%d", &money);  
printf("50000원 권 : %d개, ", money / 50000);  
printf("10000원 권 : %d개, ", (money % 50000) / 10000);  
printf("5000원 권 : %d개, ", ((money % 50000) % 10000) / 5000);  
printf("1000원 권 : %d개", (((money % 50000) % 10000) % 5000) / 1000);  
return 0;  
}
```

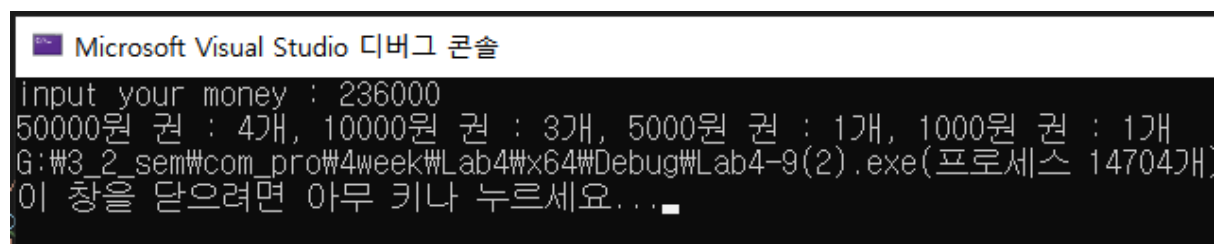
#### 4) 실행결과

[1]



```
Microsoft Visual Studio 디버그 콘솔  
(-2.550000 1.150000)  
G:\#3_2_sem\com_pro\4week\Lab4\64\Debug\Lab4-9(1).  
이 창을 닫으려면 아무 키나 누르세요...
```

[2]



```
Microsoft Visual Studio 디버그 콘솔  
input your money : 236000  
50000원 권 : 4개, 10000원 권 : 3개, 5000원 권 : 1개, 1000원 권 : 1개  
G:\#3_2_sem\com_pro\4week\Lab4\64\Debug\Lab4-9(2).exe(프로세스 14704개)  
이 창을 닫으려면 아무 키나 누르세요...
```

#### 5) 검토의견

- 조건문 또는 조건연산자를 다중으로 사용한다면 필요없는 화폐단위는 출력하지 않을 수 있을 것이라 생각. 1000원 권부터 출력하는 프로그램을 작성하여 위로 올라가는 구조로 작성하면 수월하게 코드를 작성할 수 있을 것이라 생각한다.