



1. [대입과 산술 연산자]

```
#include <stdio.h>

int main(void)
{
    int x = 3;
    int y = 2;
    int z;

    z = x + y;
    z = x - y;
    z = y - x;
    z = x * y;
    z = x / y;
    z = x % y;

    return 0;
}
```

실행결과(기록하기)

- (1) 문장이 실행될 때마다 변수 z의 값을 출력하도록 수정하라. 실행 결과를 예측하고 실제 실행 결과와 비교하여 보라. 나눗셈의 결과를 설명하여 보라.
- (2) 사용자로부터 x, y 값을 입력받도록 수정하여 보라.
- (3) 변수 x, y, z의 타입을 float로 변경하여서 다시 컴파일하고 실행하여 보라. 실행 결과가 달라지는 것은 무엇인가?

2 [복합 대입 연산자]

```
#include <stdio.h>

int main(void)
{
    int x = 5;
    int y = 2;

    x += 2;
    y *= 10;
    y /= x + 1;
    x %= x - y;
}
```



```
    return 0;
}
```

실행결과(기록하기)

- (1) 문장이 실행될 때마다 각 변수들의 값을 출력하도록 수정하라. 실행 결과를 예측하고 실제 실행 결과와 비교하여 보라.
- (2) 위의 프로그램에서 복합 대입 연산자를 대입 연산자와 산술 연산자로 풀어서 다시 작성하여 보라. 실행 결과를 (1)과 비교하여 보라.
- (3) $x += 2$ 를 $x =+ 2$ 로 변경하였을 경우, 실행 결과가 어떻게 되는가?

3 [증감 연산자]

```
#include <stdio.h>

int main(void)
{
    int x = 1;

    printf("%d\n", x++);
    printf("%d\n", x);
    printf("%d\n", ++x);
    printf("%d\n", x--);
    printf("%d\n", x);
    printf("%d\n", --x);

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램의 실행 결과를 예측하고 실제 실행 결과와 비교하여 보라.
- (2) 증감 연산자를 이항 연산자로 풀어서 다시 작성하여 보라.
- (3) $10++$, $(x+y)++$, $(x++)++$ 등이 가능한가? 직접 컴파일하여 보고 그 이유를 설명하라.

4 [형변환]

```
#include <stdio.h>

int main(void)
{
    int i, j;
    float f;

    i = 2.3 + 2.8;
    j = (int)2.3 + (int)2.8;

    f = i / j;
    f = (float)i / j;
    f = i / (float)j;
    f = (int)(i / j);
    f = (float)(i / j);

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램에서 문장이 실행될 때마다 변수 x, y, f의 값이 어떻게 변화되는지를 예측하고, 디버거나 printf() 문장을 이용하여 변수의 값을 확인하라.
- (2) 각 문장에서 이루어지는 형변환을 설명하고 내림 변환과 올림 변환을 지적하여 보라.

5 [관계연산자]

```
#include <stdio.h>

int main(void)
{
    int x = 1;

    printf("%d\n", 1 / 2 == 2 / 3);
    printf("%d\n", (2 == 2) + (2 != 3));
    printf("%d\n", 2 + 3 == 5);
    printf("%d\n", 2 != 1 + 1);
    printf("%d\n", 2 + 3 < 5);
    printf("%d\n", (1e32 + 0.01) > 1e32); // ①
    printf("%d\n", (20.12 - 12.02) == 8.1); // ②

    return 0;
}
```



실행결과(기록하기)

- (1) 위의 프로그램의 실행 결과를 예측하고 실제 실행 결과와 비교하여 보라.
- (2) 문장 ①을 설명하여 보라.
- (3) 문장 ②를 설명하여 보라.

6 [논리연산자]

```
#include <stdio.h>

int main(void)
{
    int x = 0;

    printf("%d\n", !0);
    printf("%d\n", !1);
    printf("%d\n", !!2);
    printf("%d\n", !(1 == 2) + (1 == 1));
    printf("%d\n", (1 == 2) && (1 == 1));
    printf("%d\n", (1 == 2) || (1 == 1));
    printf("%d\n", (x != 0) || (x != 1)); // ①

    ( 3 > 2 ) || ( ++x < 5 ); // ②
    printf("x = %d\n", x);

    ( x != 0 ) && ( y / x == 3 ); // ③

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램의 실행 결과를 예측하고 실제 실행 결과와 비교하여 보라.
- (2) ①번 문장을 AND 연산자로 바꾸어 보라.
- (3) 문장 ②를 설명하여 보라. x가 증가되지 않는 이유는 무엇인가?
- (4) 문장 ③에서 x가 0임에도 불구하고 y / x 연산에서 오류가 발생하지 않는 이유는 무엇인가?

7 [조건, 콤마 연산자]

```
#include <stdio.h>

int main(void)
{
    int x = 10;
    int y = 20;

    printf("%d\n", (x > y) ? x : y);    // ①
    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램의 실행 결과를 예측하고 실제 실행 결과와 비교하여 보라.
- (2) 문장 ①을 최소값을 구하도록 변경하여 보라.
- (3) 문장 ①을 변수의 x의 절대값을 구하도록 변경하여 보라.
- (4) x가 짝수이면 화면에 “짝수”라고 출력하고 x가 홀수이면 “홀수”라고 출력하는 문장을 조건 연산자를 이용하여 작성하여 보라.
- (5) 콤마 연산자를 이용하여 하나의 문장으로 변수 x에 10을, 변수 y에 20을 대입하도록 해보라.

8 [sizeof 연산자]

```
#include <stdio.h>

int main(void)
{
    char c;
    short s;
    int i;
    float f;
    double d;

}
```

실행결과(기록하기)

- (1) 위의 프로그램에 변수 c, s, i, f, d의 크기를 sizeof 연산자로 계산하여 출력하는 문장을 추가하라.
- (2) 변수 이름 대신에 변수의 자료형을 사용하여 각 자료형의 크기를 출력해보라. 즉 sizeof(i) 대신에 sizeof(int)을 사용한다.



9 [우선 순위와 결합 법칙]

```
#include <stdio.h>

int main(void)
{
    int x = 10, y;

    printf("%d\n", y = x / 3 * 2 % 6);
    printf("%d\n", y = 3 % 5 * (3 / 5, x = 3));

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램을 분석하여 실행결과를 예측하여 보라.
- (2) 위의 프로그램을 컴파일, 실행하여 예측된 실행 결과와 비교하여 보라. 디버거로 변수 x와 y의 값을 추적하여 보라.

10 [디버깅 문제]

화씨 온도를 섭씨 온도로 바꾸는 프로그램을 작성하여 보자. 변환식은 다음과 같다.

$$\text{섭씨온도} = \frac{5}{9} (\text{화씨온도} - 32)$$

```
#include <stdio.h>

int main(void)
{
    int f_temp = 60;
    int c_temp;

    c_temp = 5 / 9 * (f_temp - 32);
    printf("섭씨온도는 %d입니다.\n", c_temp);

    return 0;
}
```

- (1) 우선 위의 프로그램을 검토하여 보자. 컴파일하고 실행하여 보자. 실행 결과가 올바르게 나오는가? 만약 실행 결과가 올바르지 않다면 그 원인은 무엇인가? 디버거를 사용하여도 좋다.
- (2) 결과가 제대로 나오도록 위의 프로그램을 수정하여 보라.
- (3) 화씨 온도를 사용자로부터 받게끔 수정하라.

화씨 온도를 입력하시오: 32
섭씨 온도는 0도입니다.

- (4) 화씨 온도와 섭씨 온도를 실수로 표현하여 처리하도록 프로그램을 수정하라.
- (5) 이 프로그램과는 반대로 섭씨 온도를 입력하여 화씨 온도로 변환하는 프로그램을 작성하여 보라.