

# 1. 다양한 진법 정수 리터럴

## 1. 기능

- 10진수, 8진수, 16진수 형태의 상수를 각각 선언하고, 세 상수를 더한 결과를 10진수로 출력한다.
- 추가로, 더한 결과를 16진수, 8진수 형태로도 출력해볼 것.

## 2. 요구사항

- 예시: 10진수 10, 16진수 0x10, 8진수 010을 변수로 선언 후 출력.
- printf 형식 지정자를 사용하여 10진수(%d), 16진수(%x), 8진수(%o)로 결과를 표시.

10진수: 10 (dec), 8진수: 010 (oct), 16진수: 0x10 (hex)  
합계 (10진수) : 42  
합계 (16진수) : 2a  
합계 (8진수) : 52

# 1. 다양한 진법 정수 리터럴

```
/*
파일명: ex1.c
작성자: 홍길동
설명  : 10진수, 8진수, 16진수 상수를 선언하고 연산 결과를 출력
*/

#include <stdio.h>

int main(void)
{
    // 10진수, 16진수, 8진수 상수 예시
    int dec = 10;    // 10진수
    int hex = 0x10;  // 16진수(앞에 0x)
    int oct = 010;   // 8진수(앞에 0)

    // 합계 계산 (10진수 연산)
    int sum = dec + hex + oct;

    // 출력
    printf("10진수 dec = %d\n", dec);
    printf("16진수 hex = %x (10진수로 %d)\n", hex, hex);
    printf("8진수 oct = %o (10진수로 %d)\n", oct, oct);

    printf("\n=== 합계 계산 ===\n");
    printf("합계 (10진수) = %d\n", sum);
    printf("합계 (16진수) = %x\n", sum);
    printf("합계 (8진수) = %o\n", sum);

    return 0;
}
```

## 2. 지수 표기법과 정밀도 비교

### 1. 기능

- float 변수를 하나, double 변수를 하나 선언한다.
- 실수 리터럴을 지수 표기법(e, E)을 포함하여 여러 가지 형태(예: 3.14, 3.14e1, 2.71828e-2)로 입력하고 각각을 출력해 본다.

### 2. 요구사항

- 같은 실수 값을 float와 double에 저장할 때, 소수점 이하 자리수가 어떻게 달라지는지 출력해볼 것.
- %.4f, %.6f, %e 등 다양한 형식 지정자를 사용해보자.

```
float fval1 = 3.14e1 -> fval1 = 31.4000  
double dval1 = 3.14e1 -> dval1 = 31.400000  
float fval2 = 2.71828e-2 -> fval2 = 0.027182  
double dval2 = 2.71828e-2 -> dval2 = 0.027182
```

## 2. 지수 표기법과 정밀도 비교

```
/*
파일명: ex2.c
작성자: 홍길동
설명 : 지수 표기법을 사용한 float, double 변수를 비교하며 출력
*/

#include <stdio.h>

int main(void)
{
    // float, double 변수를 선언하고
    // 지수 표기법(e, E)을 포함한 다양한 실수 리터럴 대입
    float fval1 = 3.14e1f;      // 3.14 × 10^1 => 31.4
    float fval2 = 2.71828e-2f;  // 2.71828 × 10^-2 => 0.0271828...

    double dval1 = 3.14e1;      // 3.14 × 10^1 => 31.4
    double dval2 = 2.71828e-2;  // 2.71828 × 10^-2 => 0.0271828...

    // 소수점 이하 6자리, 8자리 등 여러 포맷으로 출력
    printf("=== float ===\n");
    printf("fval1 = %.4f, fval2 = %.8f\n", fval1, fval2);

    printf("\n=== double ===\n");
    printf("dval1 = %.4f, dval2 = %.8f\n", dval1, dval2);

    // 지수 표기법(%e) 출력도 시도
    printf("\n=== 지수 표기법으로 출력 ===\n");
    printf("fval1 = %e, dval1 = %E\n", fval1, dval1);

    return 0;
}
```

### 3. 심볼릭 상수를 이용한 도형 계산

#### 1. 기능

- 원의 넓이와 둘레를 구하는 프로그램을 작성한다.
- 심볼릭 상수로 원주율(파이,  $\pi$ )을 정의하고(const나 #define 중 하나 필수), 반지름을 입력받아 그에 맞는 넓이와 둘레를 계산하여 출력한다.

#### 2. 요구사항

- `const double PI = 3.14159;` 또는 `#define PI 3.14159` 중 적어도 하나 사용.
- 반지름(radius)은 double로 선언하거나, scanf로 사용자 입력을 받아서 저장.
- 원의 넓이 =  $PI \times (radius)^2$
- 원의 둘레 =  $2 \times PI \times radius$

반지름을 입력하세요: 5.0  
원의 넓이: 78.54  
원의 둘레: 31.42

### 3. 심볼릭 상수를 이용한 도형 계산

```
/*
파일명: ex3_const.c
작성자: 홍길동
설명 : const 키워드를 사용한 원의 넓이, 둘레 계산
*/

#include <stdio.h>

int main(void)
{
    const double PI = 3.14159; // 심볼릭 상수 (변경 불가)
    double radius; // 반지름

    printf("반지름을 입력하세요: ");
    scanf("%lf", &radius);

    double area = PI * radius * radius; // 넓이
    double circumference = 2.0 * PI * radius; // 둘레

    printf("원의 넓이: %.2f\n", area);
    printf("원의 둘레: %.2f\n", circumference);

    return 0;
}
```

### 3. 심볼릭 상수를 이용한 도형 계산

```
/*
파일명: ex3_define.c
작성자: 홍길동
설명 : #define 전처리를 사용한 원의 넓이, 둘레 계산
*/

#include <stdio.h>
#define PI 3.14159 // 심볼릭 상수 매크로

int main(void)
{
    double radius; // 반지름

    printf("반지름을 입력하세요: ");
    scanf("%lf", &radius);

    double area = PI * radius * radius;
    double circumference = 2.0 * PI * radius;

    printf("원의 넓이: %.2f\n", area);
    printf("원의 둘레: %.2f\n", circumference);

    return 0;
}
```