



실 습 문 제

제4장

1. [변수의 선언과 초기화]

다음 표와 같은 초기값을 가지는 변수들을 선언하고 초기화하라. 먼저 각 변수의 값을 적절하게 표현할 수 있는 자료형을 선택하여 변수를 선언한다. 다음으로 적절한 형식 지정자를 사용한 printf()를 이용하여 화면에 변수의 값을 출력하라.

자료형	변수이름	초기값
	grade	'A'
	weight	78kg
	salary	2,000,000원
	distance1	149,600,000km
	price_of_apartment	2,200,000,000원
	height	178.9
	distance2	2×10^{19} km
	distance3	3×10^{123} km

```
#include <stdio.h>

int main(void)
{
    _____; // 변수 grade를 선언하고 초기화한다.
    _____; // 변수 weight를 선언하고 초기화한다.
    _____; // 변수 salary를 선언하고 초기화한다.
    _____; // 변수 distance1을 선언하고 초기화한다.
    _____; // 변수 price_of_apartment를 선언하고 초기화한다.
    _____; // 변수 height를 선언하고 초기화한다.
    _____; // 변수 distance2를 선언하고 초기화한다.
    _____; // 변수 distance3를 선언하고 초기화한다.

    printf("_____"); // 변수 grade를 출력한다.
    printf("_____"); // 변수 weight를 출력한다.
    printf("_____"); // 변수 salary를 출력한다.
    printf("_____"); // 변수 distance1를 출력한다.
    printf("_____"); // 변수 price_of_apartment를 출력한다.
    printf("_____"); // 변수 height를 출력한다.
    printf("_____"); // 변수 distance2를 출력한다.
    printf("_____"); // 변수 distance3를 출력한다.

    return 0;
}
```




3. [실수형 변수와 상수에 대한 실습]

```
#include <stdio.h>

int main(void)
{
    float x;

    x = 123.4567;
    printf("x = %f\n", x);

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램을 오류없이 입력하고 컴파일하여 실행하라.
- (2) 실수 상수 123.4567을 지수 표기법으로 표기하여 보라.
- (3) 형식 지정자 %f 대신에 %e를 사용하여 출력하여 보라. %g로 표기하면 어떤 실행 결과가 나오는가?
- (4) 123.4567은 실수 상수이다. 기본적으로 실수 상수는 몇 바이트로 표현되는가? 이것을 알아보기 위하여 다음의 문장을 추가하여 보라.

```
printf("실수 상수의 크기 = %d \n", sizeof(123.4567));
```

- (5) 123.4567을 4 바이트의 실수 상수로 만들려면 어떻게 하여야 하는가? sizeof 연산자로 그 크기를 확인하여 보라.
- (6) 변수 x에 0.1234567890123456789를 대입하고 출력하여 보라. 출력 결과를 설명하여 보라.
- (7) 변수 x에 1e39를 대입하여 보자. 출력 결과를 설명하여 보라.
- (8) 변수 x에 1.23456e-46을 대입하여 보자. 출력 결과를 설명하여 보라.
- (9) 사용자로부터 하나의 실수를 입력받아서 변수 x에 저장하여 보라.
- (10) double형의 변수 y를 선언하여 (6)번을 다시 실습해보라. double형의 정밀도는 소수점 이하 몇 자리라고 할 수 있는가?
- (11) 사용자로부터 하나의 실수를 입력받아서 double형의 변수 y에 저장하여 보라. %f 형식 지정자를 사용하는 경우, 제대로 입력되는가? %lf 형식 지정자를 사용할 때는 어떤가?
- (12) 사용자로부터 하나의 실수를 입력받아서 double형의 변수 y에 저장하여 보라. 실수를 입력할 때 지수형태로 입력하여 보라. 예를 들면 0.123456e+10을 입력하여 보라.

4. [문자형 변수와 상수에 대한 실습]

```
#include <stdio.h>

int main(void)
{
    char c = 0;           // char형 변수 c 선언 및 초기화

    c = 'A';             // c에 문자 'A' 대입
    printf("c = %c\n", c); // c를 %c 형식으로 출력

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램을 오류없이 입력하고 컴파일하여 실행하라. 어떤 결과가 출력되는가?
- (2) 문자 'A'가 저장되어 있는 c를 %d 형식지정자를 이용하여 출력하여 본다. 어떤 값이 나오는가? C언어에서 문자는 어떤 값으로 저장되는가? 이 값은 무엇을 의미하는가?
- (3) 변수 c에 'A'에 저장되어 있는 상태에서 c, c-1, c+1을 %c 형식과 %d 형식으로 각각 출력하여 보라. 어떤 값이 출력되는가? 이것은 무엇을 의미하는가?
- (4) '\101', 0x41, 65, 'A'을 형식 지정자 %c를 이용하여 출력하여 보라. 실행 결과를 설명하여 보라.

```
printf("%c %c %c %c", '\101', 0x41, 65, 'A');
```

- (5) 7과 '7'을 형식 지정자 %d를 이용하여 출력하여 보라. 차이점은 무엇인가?

```
printf("%d %d", 7, '7');
```

- (6) c에 127, 128, 129, 255, 256을 차례대로 대입해서 %d 형식 지정자를 이용하여 출력하여 본다. 어떤 결론을 내릴 수 있는가? 어떤 범위에서 오버플로우(overflow, 넘침)가 발생하는가? 오버플로우가 발생했을 경우, 저장되는 값은 어떤 값인가?
- (7) unsigned char형으로 변수 uc를 선언한다. uc에 127, 128, 129, 255, 256을 차례대로 대입해서 %d 형식 지정자를 이용하여 출력하여 본다. char형과 unsigned char형의 차이점은 무엇인가?
- (8) 사용자로부터 문자 하나를 입력받아서 변수 c에 저장하여 보라. 어떤 형식 지정자를 사용하여야 하는가?



5. [기호 상수]

```
#include <stdio.h>
#define PI 3.14

int main(void)
{
    float radius;
    float peri, area;

    printf("원의반지름을입력하시오: ");
    scanf("%f", &radius);
    peri = 2 * PI * radius;
    area = PI * radius * radius;

    printf("원의둘레는%f입니다.\n", peri);
    printf("원의면적은%f입니다.\n", area);

    return 0;
}
```

실행결과(기록하기)

- (1) 위의 프로그램을 오류없이 입력하고 컴파일하여 실행하라.
- (2) 원주율을 3.14에서 3.141592로 변경하여 보자. 어디를 변경하면 되는가?
- (3) 위의 프로그램을 #define을 사용하지 말고 다시 작성하여 보라. 원주율을 3.14에서 3.141592로 변경하여 보자. 몇 군데나 변경하여야 하는가?
- (4) 만약 main() 함수의 첫 줄에 다음 문장을 두어 PI 값을 바꾸면 어떻게 되는가?

```
PI = 3.141592;
```

- (5) {와 }을 BEGIN과 END로 재정의를 하여 위의 프로그램을 다시 작성하여 보라. 이것은 파스칼이라는 언어에서 사용되던 키워드이다.

```
#define BEGIN {
#define END }
```

- (6) 본문을 참조하여 PI를 const 키워드를 사용하여 다시 정의하여 보라.