

기초프로그래밍

제2장 c 언어의 기본 구조와 표준 입출력

Sangsoo Lim

CSAI

Dongguk University

차례

- C언어의 기본 구조
- Hello C world 프로그램
- printf() 함수
- scanf() 함수

C언어의 기본 구조

Step1

```
/*  
    파일명: Hello.c  
    만든이: 홍길동  
    프로그램 내용: Hello C world 테스트  
*/
```

Step2

```
→ #include<stdio.h>
```

Step3

```
{  
    int main(void)  
    {  
        printf("Hello C world \n");  
        return 0;  
    }  
}
```

C언어의 기본 구조

- Step1 – 주석

- ① 주석(Comment): 메모의 기능
- ② 컴파일러는 주석의 내용을 무시

- ③ 주석 처리 방법

```
/*  
파일명: Hello.c  
만든이: 홍길동  
프로그램 내용: Hello C world 테스트  
*/
```

여러 줄 주석 처리

```
// 파일명: Hello.c  
// 만든이: 홍길동  
// 프로그램 내용: Hello C world 테스트
```

한 줄 주석 처리

C언어의 기본 구조

- Step1 – 주석

④ 주석 처리 시 주의 사항 – 중복 사용 금지

```
/*  
    /* 파일명: Hello.c  
       만드이: 홍길동  
       프로그램 내용: Hello C world 테스트 */  
*/
```

오류 발생

⑤ 주석 중복 오류 해결 방법 - // 사용

```
/*  
    // 파일명: Hello.c  
    // 만드이: 홍길동  
    // 프로그램 내용: Hello C world 테스트  
*/
```

정상

C언어의 기본 구조

• Step1 – 주석

```
/*
    파일명 : Hello.c
    /* 만든이 : 홍길동 */
    프로그램 내용 : Hello C world 테스트
*/

#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```



/* /* */ */
주석 중복
오류 발생

```
/*
    파일명 : Hello.c
    // 만든이 : 홍길동
    프로그램 내용 : Hello C world 테스트
*/

#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```



/* // */
주석 중복
오류 해결

C언어의 기본 구조

Step 1

```
/*  
    파일명: Hello.c  
    만든이: 홍길동  
    프로그램 내용: Hello C world 테스트  
*/
```

Step 2

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

Step 3

```
int main(void)  
{  
    printf("Hello C world \n");  
    return 0;  
}
```

C언어의 기본 구조

- Step2 – 전처리기와 헤더파일

① ② ③
`#include <stdio.h>`

① 전처리기(preprocessor)- # : 컴파일을 수행하기 전에 먼저 처리

② `include` : 전처리 동작 규정

③ 헤더파일(Header File) : 확장자 .h를 가지는 파일

- `stdio` 의미 : **S**tandard **I**nput **O**utput (표준 입력 출력)

- `stdio.h` 의미 : **표준 입력 출력** 함수들을 가지고 있는 헤더 파일

C언어의 기본 구조

Step 1

```
/*  
    파일명: Hello.c  
    만든이: 홍길동  
    프로그램 내용: Hello C world 테스트  
*/
```

Step 2

```
→ #include<stdio.h>
```

Step 3

```
{  
    int main(void)  
    {  
        printf("Hello C world \n");  
        return 0;  
    }  
}
```

C언어의 기본 구조

- Step3 – main() 함수

① main() 함수의 호출과 종료는 누가 결정 하는가?

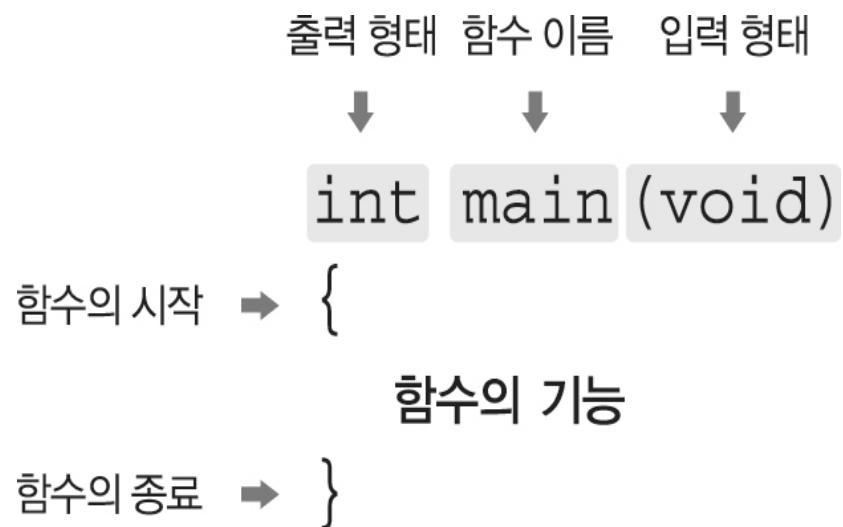
- 운영체제에 의해서 호출



```
int main (void)
{
    printf ("Hello C world \n");
    return 0;
}
```

C언어의 기본 구조

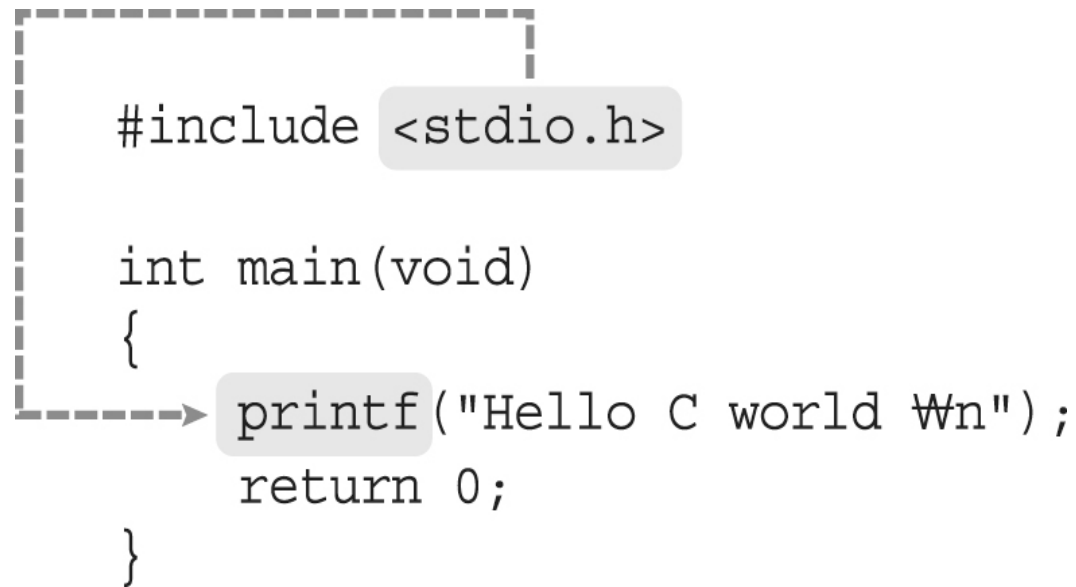
- Step3 – main() 함수



C언어의 기본 구조

- Step3 – main() 함수

② 표준 라이브러리 함수와 헤더파일은 프로그래밍을 편하게 한다.



```
#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```

A diagram illustrating the relationship between the `#include <stdio.h>` directive and the `printf` function. A dashed line connects the `<stdio.h>` header file to the `printf` function call within the `main` function, indicating that the function is defined in the included header file.

C언어의 기본 구조

- Step3 – main() 함수

③ 세미콜론(;) 은 문장의 끝을 의미

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

```
int main(void)
```

```
{
```

```
    printf("Hello C world \n");
```

```
    return 0;
```

```
}
```

세미콜론

C언어의 기본 구조

- Step3 – main() 함수

④ return은 반환과 종료의 의미를 갖는다.

```
#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return 0;
}
```

운영체제에게 0을 반환 하고,
main() 함수를 종료해라!

```
#include <stdio.h>

int main(void)
{
    printf("Hello C world \n");
    return;
}
```

반환할 것이 없으니 main()
함수를 그냥 종료해라!

Hello C world

// 예제 2-1.c

```
#include <stdio.h>    // 전처리하는 부분
int main(void)        // 운영체제가 호출
{
    printf("Hello C world"); // 모니터에 출력

    return 0;          // 운영체제에게 값 반환 후, main() 종료
}
```

모니터에 데이터 출력하기 – printf() 함수

// 예제 2-1.c

```
#include <stdio.h>
int main(void)
{
    printf("Hello C world");

    return 0;
}
```

// 예제 2-2.c

```
#include <stdio.h>
int main(void)
{
    printf("Hello C world \n");

    return 0;
}
```


모니터에 데이터 출력하기 – printf()

- 특수 문자

특수 문자	설명
\a	경고음 소리 발생
\b	백스페이스(Backspace)
\f	폼 피드(Form Feed)
\n	개행(New Line)
\r	캐리지 리턴(Carriage Return)
\t	수평 탭
\v	수직 탭
\\	역슬래시(₩)
\'	작은따옴표
\"	큰따옴표

모니터에 데이터 출력하기 – printf()

- 특수 문자 예제 실습 : 2-3.c

```
#include <stdio.h>
int main(void)
{
    printf("\t Hello C! \n Hello World! \n");
    printf("큰따옴표: \" \" \n");
    printf("작은따옴표: \' \' \n");
    printf("역슬레시: \\ \n");

    return 0;
}
```

모니터에 데이터 출력하기 – printf()

- 서식 문자 (서식화된 문자)
 - 출력 서식 문자

```
printf(“ %d ”, 3+5);
```

A dashed green arrow points from the expression '3+5' in the second argument of the printf function to the format specifier '%d' in the first argument.

3+5의 결과를 출력 서식 문자 %d로...

모니터에 데이터 출력하기 – printf()

- 출력 서식 문자 (서식화된 출력 문자)

서식문자	출력 형태
%d, %i	10진수 정수 (양수와 음수 모두 표현 가능)
%x, %o	16진수 정수, 8진수 정수 (양수만 표현 가능)
%f, %lf	10진수 실수 (양수와 음수 모두 표현 가능)
%c	한 개의 문자
%s	문자열
%u	10진수 정수 (양수만 표현 가능)
%e	e 표기법에 의한 실수
%E	E 표기법에 의한 실수
%g	소수점 이하 자리 수에 따라 %f, %e 둘 중 하나를 선택
%G	소수점 이하 자리 수에 따라 %f, %E 둘 중 하나를 선택
%%	% 기호 출력

모니터에 데이터 출력하기 – printf()

- 출력 서식 문자 (서식화된 출력 문자)

① 숫자 출력하기 - %d, %i, %x, %o, %f, %lf

서식문자	출력 형태
%d, %i	10진수 정수 (양수와 음수 모두 표현 가능)
%x, %o	16진수 정수, 8진수 정수 (양수만 표현 가능)
%f, %lf	10진수 실수 (양수와 음수 모두 표현 가능)

모니터에 데이터 출력하기 – printf()

- 숫자 출력하기 예제 실습 : 2-6.c

```
#include <stdio.h>
int main(void)
{
    printf("10진수 정수: %d \n", 0.5);
    printf("10진수 실수: %f \n", 0.5);
    printf("10진수 실수: %1f \n", 0.5);

    printf("소수점 이하 6자리이상: %f \n", 0.5655678);
    printf("소수점 이하 6자리이상: %1f \n", 0.5667784);

    return 0;
}
```

```
10진수 정수: 0
10진수 실수: 0.500000
10진수 실수: 0.500000
소수점 이하 6자리이상: 0.565568
소수점 이하 6자리이상: 0.566778
```

모니터에 데이터 출력하기 – printf()

- 출력 서식 문자 (서식화된 출력 문자)

② 문자와 문자열 출력하기- %c, %s

서식문자	출력 형태
%c	한 개의 문자
%s	문자열

모니터에 데이터 출력하기 – printf()

- 문자와 문자열 출력하기 예제 실습 : 2-7.c

```
#include <stdio.h>
int main(void)
{
    printf("a를 대문자로 표현하면 %c 입니다. \n", 'A');

    printf("%s %c 입니다. \n", "a를 대문자로 표현하면", 'A');

    printf("%s %d %s \n", "2곱하기3은", 2*3, "입니다.");

    return 0;
}
```


모니터에 데이터 출력하기 – printf()

- 출력 서식 문자 (서식화된 출력 문자)

④ 기타 실수 표기법 - %e, %E, %g, %G, %%

서식문자	출력 형태
%e	e 표기법에 의한 실수
%E	E 표기법에 의한 실수
%g	소수점 이하 자리 수에 따라 %f, %e 둘 중 하나를 선택
%G	소수점 이하 자리 수에 따라 %f, %E 둘 중 하나를 선택
%%	% 기호 출력

모니터에 데이터 출력하기 – printf()

- 기타 실수 표기법 예제 실습 : 2-9.c

```
#include <stdio.h>
int main(void)
{
    printf("%f \n", 0.000123);          // 소수점 이하 6자리
    printf("%f \n", 0.0001236);         // 소수점 이하 6자리 초과

    printf("%e \n", 0.000123);          // 소수점 이하 6자리
    printf("%E \n", 0.0001236);         // 소수점 이하 6자리 초과

    printf("%g \n", 0.000123);          // 소수점 이하 6자리
    printf("%G \n", 0.0000123456);      // 소수점 이하 6자리 초과

    printf("올해 우리나라 경제 성장률은 5%% 다. \n");
    return 0;
}
```

0.000123
0.000124
1.230000e-04
1.236000E-04
0.000123
1.23456E-05
올해 우리나라 경제 성장률은 5% 다.

모니터에 데이터 출력하기 – printf()

• 필드 폭 지정

- **%03d** : 필드 폭을 3칸 확보하고 오른쪽 정렬, 남은 자리는 0으로 채움
- **%-3d** : 필드 폭을 3칸 확보하고 왼쪽 정렬
- **%+3d** : 필드 폭 3칸 확보하고 오른쪽 정렬, 양수는 +, 음수는 - 출력

```
#include <stdio.h>
int main(void)
{
    printf("%03d, %03d, %03d \n", 1, 20, 300);
    printf("%-3d, %-3d, %-3d \n", 1, 20, 300);
    printf(" %+3d,  %+3d,  %+3d \n", 1, 20, 300);
    printf(" %+3d,  %+3d,  %+3d \n", 1, 20, -300);

    return 0;
}
```

```
001, 020, 300
1  , 20 , 300
+1, +20, +300
+1, +20, -300
```

키보드로부터 데이터 입력받기 – scanf()

- printf()함수 vs. scanf()함수

- printf()함수

- 모니터에 데이터를 출력하는 함수
- `stdio.h` 헤더 파일이 필요

- scanf()함수

- 키보드로 데이터를 입력하는 함수
- `stdio.h` 헤더 파일이 필요

키보드로부터 데이터 입력받기 – scanf()

- scanf() 함수의 기본 구조
 - 입력 서식 문자
 - 변수 (데이터를 저장하는 임시 공간)

scanf("%d", &a);

입력 서식 문자

입력을 저장하는 변수



- ① 키보드로부터 데이터를 입력 서식 문자 %d 형식으로 입력 받는다.
- ② 입력받은 데이터를 변수 a에 저장한다.

키보드로부터 데이터 입력받기 – scanf()

- scanf() 함수 예제 실습 : 2-11.c

```
#include <stdio.h>
int main(void)
{
    int age;
    printf("What is your age?: ");
    scanf("%d", &age);
    printf("Wow! Really? Are you %d years old?\n", age);

    return 0;
}
```

```
What is your age?: 20
Wow! Really? Are you 20 years old?
```

키보드로부터 데이터 입력받기 – scanf()

- 입력 서식 문자 (서식화된 입력 문자)

서식문자	입력 형태
%d	10진수 정수 입력
%x	16진수 정수 입력
%o	8진수 정수 입력
%f	float형 실수 입력
%lf	double형 실수 입력
%c	한 개의 문자 입력 <나중에 배울 예정>
%s	문자열 입력 <나중에 배울 예정>
%u	10진수 정수(양수만 표현 가능) 입력 <나중에 배울 예정>
%e	float형 e 표기법에 의한 실수 입력
%le	double형 e표기법에 의한 실수 입력

키보드로부터 데이터 입력받기 – scanf()

- 입력 서식 문자 (서식화된 출력 문자)

① 정수 입력 하기 - %d, %x, %o

서식문자	입력 형태
%d	10진수 정수 입력
%x	16진수 정수 입력
%o	8진수 정수 입력

키보드로부터 데이터 입력받기 – scanf()

- scanf()함수 예제 실습 : 2-12.c

```
#include <stdio.h>
int main(void)
{
    int a, b, c;

    printf("10진수 정수 1개 입력: ");
    scanf("%d", &a);
    printf("10진수: %d, 16진수: %x, 8진수: %o \n", a, a, a);

    printf("16진수 정수 1개 입력: ");
    scanf("%x", &b);
    printf("10진수: %d, 16진수: %x, 8진수: %o \n", b, b, b);

    printf("8진수 정수 1개 입력: ");
    scanf("%o", &c);
    printf("10진수: %d, 16진수: %x, 8진수: %o \n", c, c, c);
    return 0;
}
```

키보드로부터 데이터 입력받기 – scanf()

- 입력 서식 문자 (서식화된 출력 문자)

② 실수 입력 하기 - %f, %lf, %e, %le

서식문자	입력 형태
%f	float형 실수 입력
%lf	double형 실수 입력
%e	float형 e 표기법에 의한 실수 입력
%le	double형 e표기법에 의한 실수 입력

키보드로부터 데이터 입력받기 – scanf()

- scanf() 함수 예제 실습 : 2-13.c

```
#include <stdio.h>
int main(void)
{
    float f1, f2;
    double d1, d2;

    printf("float형 실수 두 개 입력: ");
    scanf("%f %e", &f1, &f2);
    printf("float형 실수 출력: f1=%f, f2=%e \n", f1, f2);

    printf("double형 실수 두 개 입력: ");
    scanf("%lf %le", &d1, &d2);
    printf("double형 실수 출력: d1=%lf, d2=%le \n", d1, d2);

    return 0;
}
```

Summary

- c언어의 기본 구조
- 모니터에 데이터를 출력하기 위해 `printf()` 함수에 사용되는 서식 문자
- 키보드로 데이터를 입력받기 위해 `scanf()` 함수에 사용되는 서식 문자
- `scanf()` 함수로 입력받은 데이터를 저장하고 활용하는 방법

1. 주석(Comment)과 기본 구조 연습

- 예제 1-1: 주석 달기와 기본 구조

```
/*
파일명: ex1_1.c
작성자: 홍길동
설명   : 주석과 기본 구조 복습
*/

#include <stdio.h> // 표준 입출력 헤더
// main() 함수 시작
int main(void)
{
    // 아래 printf 함수를 통해 문장을 출력한다.
    printf("C 언어의 기본 구조와 주석 실습!\n");

    // 프로그램을 종료하며 운영체제에게 0을 반환
    return 0;
}
```

2. printf() 함수 심화

- 예제 2-1: 다양한 특수문자 활용

```
#include <stdio.h>

int main(void)
{
    printf("\t1) 특수문자 연습\n");    // 수평 탭
    printf("2) 줄바꿈 문자는 이렇게\n사용합니다.\n");
    printf("3) 역슬래시 출력: \\n");
    printf("4) \"큰따옴표\"와 \"작은따옴표\" 출력\n");
    printf("\a5) 경고음도 낼 수 있습니다!\n");    // \a는 PC 환경에 따라
동작이 달라질 수 있음

    return 0;
}
```

2. printf() 함수 심화

- 예제 2-2: 서식문자 정렬 및 폭 지정

```
#include <stdio.h>

int main(void)
{
    // 정수 서식문자의 폭 지정 실습
    printf("[%%03d] 서식: %03d\n", 7);      // 3칸 확보, 빈자리 0으로 채움
    printf("[%%-3d] 서식: %-3d\n", 7);      // 3칸 확보, 왼쪽 정렬
    printf("[%%+3d] 서식: %+3d\n", 7);      // 3칸 확보, +, - 표시

    // 실수 서식문자 출력 폭 지정 및 소수점
    printf("[%8.2f]\n", 3.14159);           // 총 8칸 확보, 소수점 이하 2자리
    printf("[%10.4f]\n", 3.14159);         // 총 10칸 확보, 소수점 이하 4자리

    return 0;
}
```

2. printf() 함수 심화

- 예제 2-3: 여러 서식문자 혼합 사용

```
#include <stdio.h>

int main(void)
{
    int x = 255;
    double pi = 3.1415926535;

    printf("10진수: %d, 16진수: %x, 8진수: %o\n", x, x, x);
    printf("소수점 이하 2자리까지 표시: %.2f\n", pi);
    printf("지수 표기법: %e\n", pi);
    printf("g/G 자동 선택: %g / %G\n", pi, pi);

    return 0;
}
```


3. scanf() 함수 심화

- 예제 3-1: 간단한 사용자 입력 -> 출력

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("정수를 입력하세요: ");
    scanf("%d", &num);

    printf("입력된 정수: %d\n", num);
    return 0;
}
```

3. scanf() 함수 심화

- 예제 3-2: 여러 값 입력

```
#include <stdio.h>

int main(void)
{
    int a, b;
    float c;

    printf("정수 두 개와 실수 한 개를 차례대로 입력하세요:\n");
    scanf("%d %d %f", &a, &b, &c);

    printf("입력된 값: 정수1=%d, 정수2=%d, 실수=%.2f\n", a, b, c);
    return 0;
}
```

3. scanf() 함수 심화

- 예제 3-3: 16진수, 8진수 입력 후 10진수로 변환 출력

```
#include <stdio.h>

int main(void)
{
    int dec, hex, oct;

    printf("10진수로 해석할 정수 입력: ");
    scanf("%d", &dec);

    printf("16진수로 해석할 정수 입력 (예: FF): ");
    scanf("%x", &hex);

    printf("8진수로 해석할 정수 입력 (예: 377): ");
    scanf("%o", &oct);

    printf("\n==== 입력값을 10진수로 표현 =====\n");
    printf("10진수 입력: %d\n", dec);
    printf("16진수 입력: %d\n", hex);
    printf("8진수 입력: %d\n", oct);

    return 0;
}
```

3. scanf() 함수 심화

- 예제 3-4: 실수 입력 후 출력

```
#include <stdio.h>

int main(void)
{
    float f1, f2;
    double d1, d2;

    printf("float형 실수 두 개 입력 (예: 1.23 4.56): ");
    scanf("%f %f", &f1, &f2);

    printf("double형 실수 두 개 입력 (예: 7.89 1.234e-2): ");
    scanf("%lf %lf", &d1, &d2);

    printf("\n--- 입력 결과 ---\n");
    printf("float 입력: f1 = %f, f2 = %f\n", f1, f2);
    printf("double 입력: d1 = %lf, d2 = %lf\n", d1, d2);

    return 0;
}
```

1. 간단한 사칙연산 계산기

1. 프로그램 기능

- 사용자로부터 정수 두 개를 입력 받는다.
- 입력 받은 두 수에 대해 다음의 결과를 순서대로 출력한다.
 1. 덧셈(+)
 2. 뺄셈(-)
 3. 곱셈(*)
 4. 정수 나눗셈(/)
 5. 나머지(%)

2. 세부 요구사항

- 나머지 연산(%)을 출력할 때, printf에서 % 기호를 출력하려면 %%로 작성해야 함에 유의한다.
- 예를 들어, 두 수로 12와 5를 입력 받았다면 아래와 같은 결과가 출력되도록 하자.

정수 두 개를 입력하세요: 12 5

==== 사칙연산 결과 ====

$$12 + 5 = 17$$

$$12 - 5 = 7$$

$$12 * 5 = 60$$

$$12 / 5 = 2$$

$$12 \% 5 = 2$$

2. 원의 넓이와 둘레 계산

1. 프로그램 기능

- 사용자로부터 원의 반지름(실수 형태)을 입력받는다.
- 입력된 반지름을 이용하여 원의 넓이와 원의 둘레를 계산하고 결과를 출력한다.

2. 세부 요구사항

- 원주율(파이, π)은 ****상수(const)****로 선언해서 사용해보자.
- 원의 넓이: $\pi \times \text{반지름} \times \text{반지름}$
- 원의 둘레: $2 \times \pi \times \text{반지름}$
- 소수점은 적절히 조절하여 출력(예: 소수점 둘째 자리까지).

원의 반지름을 입력하세요: 2.5

반지름: 2.50

넓이 : 19.63

둘레 : 15.71

3. 간단 성적 처리

1. 프로그램 기능

- 사용자로부터 세 과목 점수(정수 형태)를 차례대로 입력 받는다. (예: 국어, 영어, 수학)
- 사용자로부터 세 과목 점수(정수 형태)를 차례대로 입력 받는다. (예: 국어, 영어, 수학)

2. 세부 요구사항

- 합계(sum)는 정수로, 평균(avg)은 실수(double)로 선언해 보자.
- 평균을 구할 때는 실수로 계산될 수 있도록, 나누는 값을 3.0과 같이 실수로 처리한다.
- 평균 출력 시 소수점 둘째 자리 정도를 출력하도록 포매팅(예: %.2f)을 시도해본다.

국어, 영어, 수학 점수를 공백으로 구분하여 입력: 95 80 88

===== 성적 결과 =====
국어: 95, 영어: 80, 수학: 88
합계: 263
평균: 87.67