



# **C 언어 프로그래밍**

Part 01. C 언어 준비 학습

## **Chapter 02. C 프로그래밍의 기초**

# 목차

1. 소스 파일과 헤더 파일이란?
2. C 코드의 기초
3. C 프로그램의 기초
4. 표준 입출력이란?
5. 프로그램 생성 및 실행

[실전예제] int와 float 사이의 오차 구하기

# 학습목표

- 소스 파일과 헤더 파일의 개념을 이해하고 직접 프로그램을 만들어본다.
- C 코드를 구성하는 기본 요소를 익힌다.
- C 프로그램의 필수 요소인 함수와 main 함수를 이해한다.
- 표준 입출력 함수인 printf, scanf 함수의 개념을 이해한다.

01

소스 파일과 헤더 파일이란?

# 01. 소스 파일과 헤더 파일이란?

## I. 프로그램이 기록되는 위치

- C 언어가 기록되는 파일

- 소스 파일(\*.c) : 확장자가 c인 파일, 프로그램의 핵심 코드가 담기는 파일
- 헤더 파일(\*.h) : 확장자가 h인 파일, 참조 파일

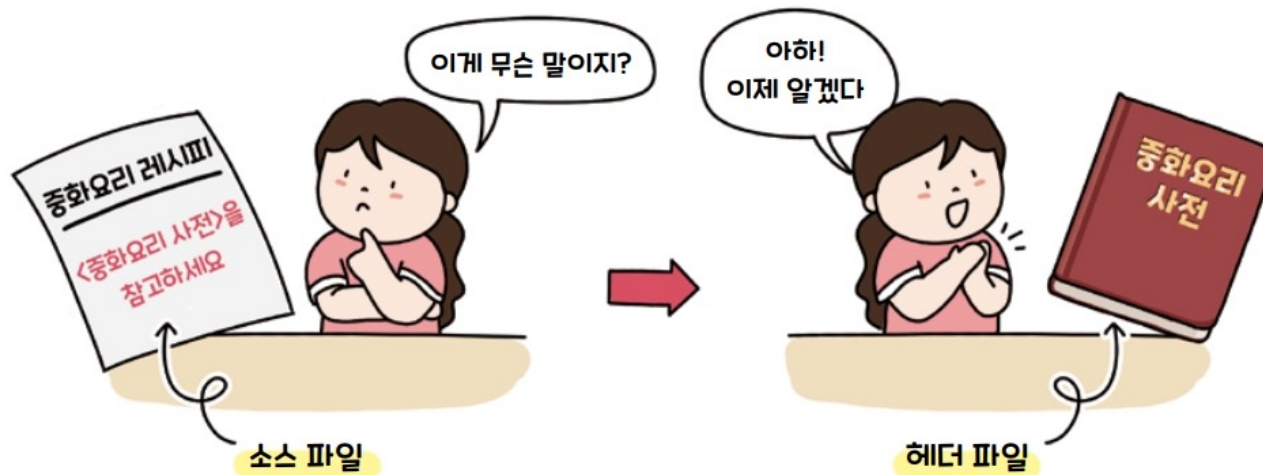
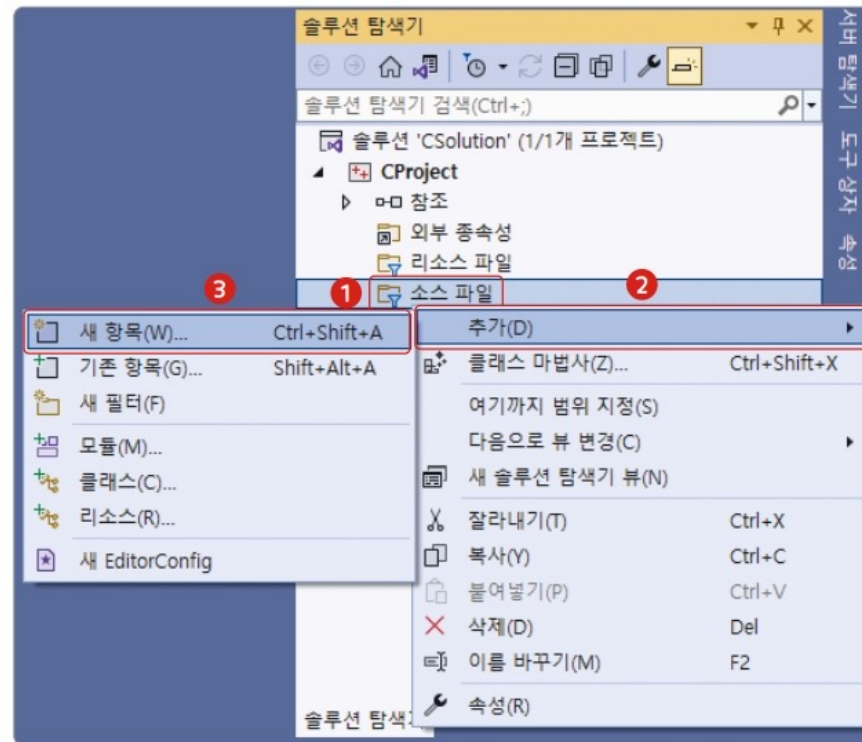


그림 2-1 소스 파일과 헤더 파일

# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

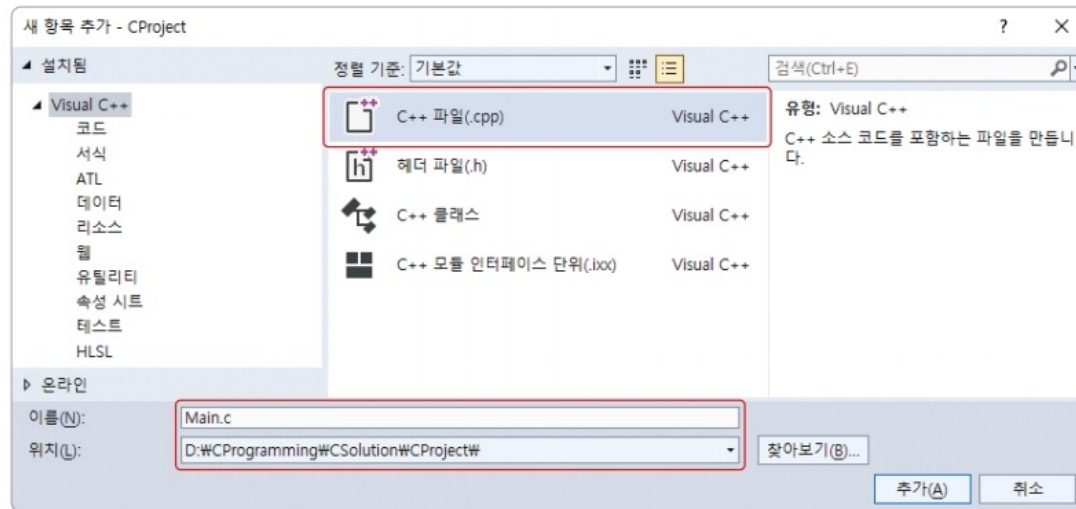
1. VS2022를 실행하고 [소스 파일] 폴더에서 마우스 오른쪽 클릭 후 [추가] [새 항목]을 선택



# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

2. [C++ 파일(.cpp)]이 선택되어 있는지 확인 후 [이름]에 Main.c를 입력



### 하나 더 알기

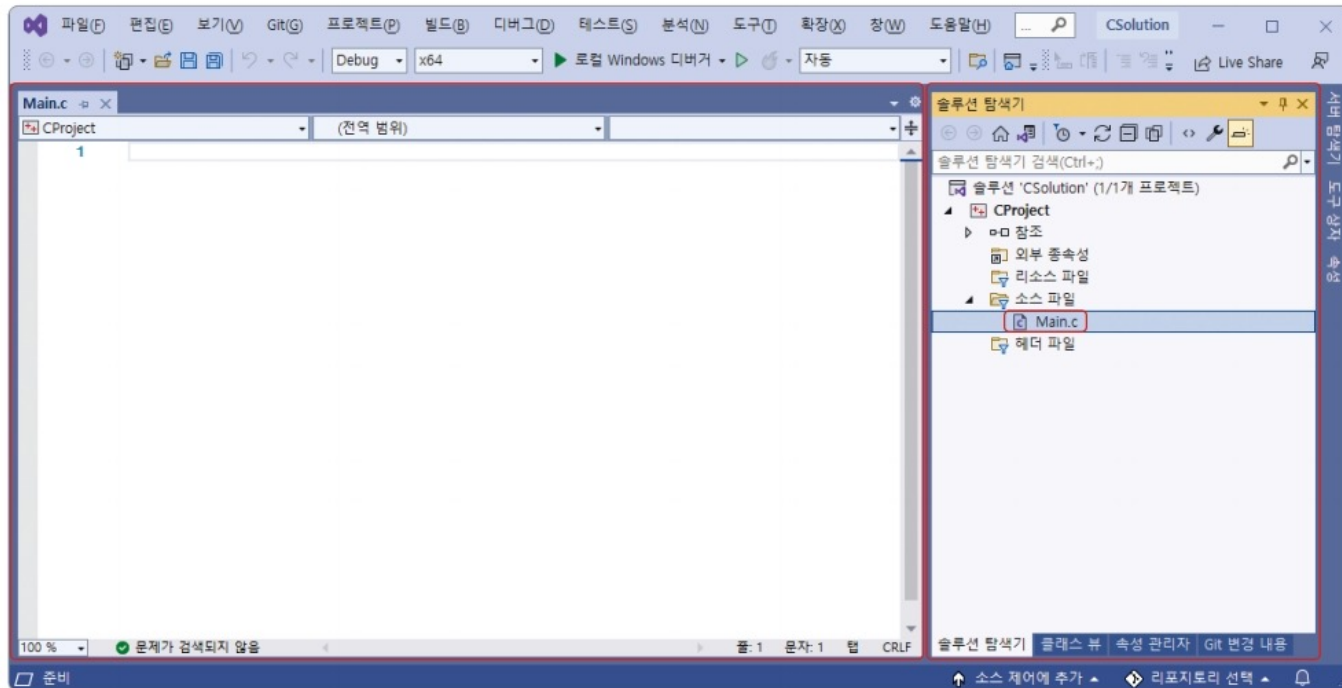
### cpp와 c 확장자

- VS2022는 확장자가 cpp인 C++의 소스 파일은 목록에서 골라서 쉽게 추가할 수 있지만, 확장자가 c인 C 언어 소스 파일은 그렇지 못함
- 순수 C 소스 파일을 작성하기 위해서는 'Main.c'와 같이 확장자를 직접 입력해야 함

# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

3. 입력을 완료하고 <추가> 버튼을 클릭하면 Main.c 파일이 추가됨





# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

### 4. 코드 편집기에 다음과 같이 코드를 입력

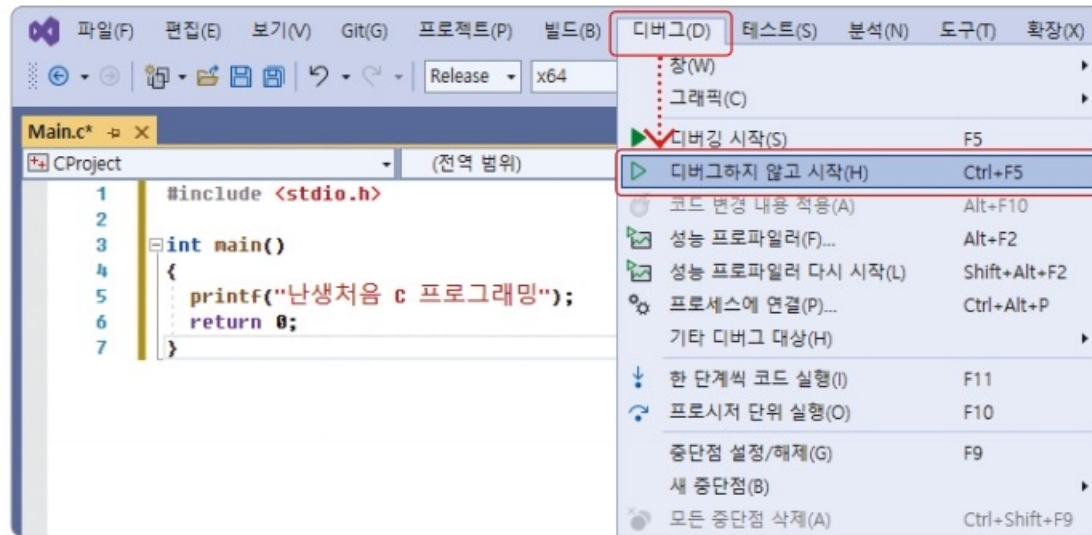
#### [코드 2-1] 최초의 프로그램

```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("난생처음 c 프로그래밍");
06      return 0;
07  }
```

# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

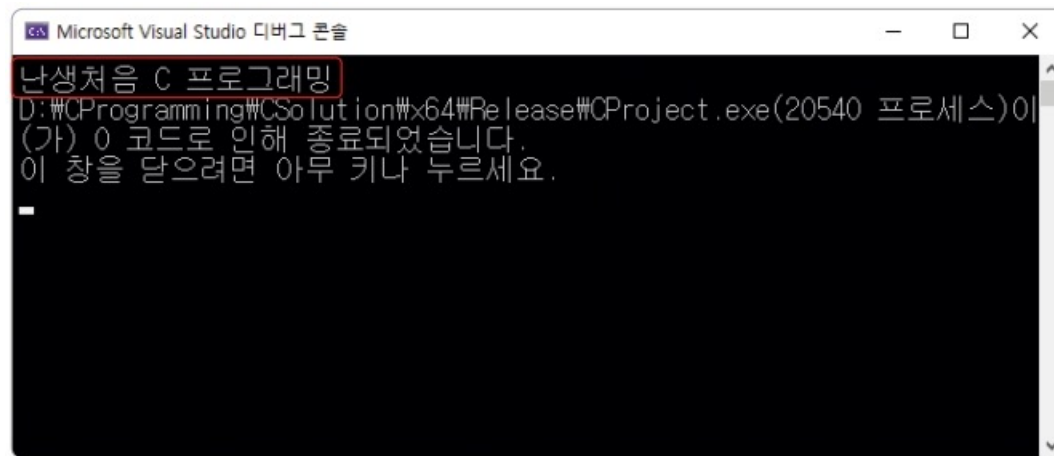
5. VS2022 메뉴에서 [디버그] [디버 그하지 않고 시작]을 선택하거나 Ctrl+F5를 눌러서 실행



# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

### 6. 다음과 같은 콘솔 창이 뜨면서 결과가 출력됨



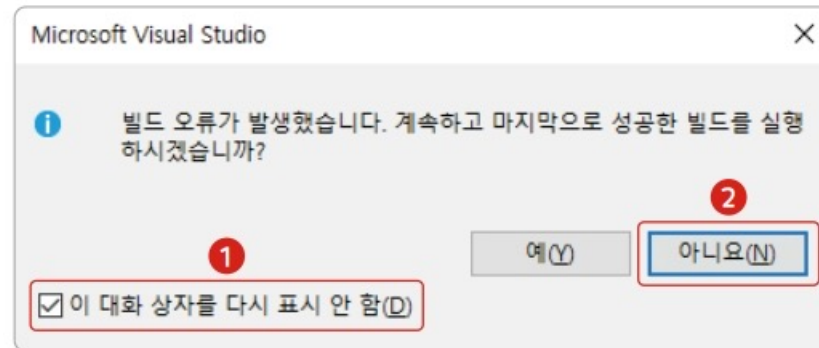
The screenshot shows a console window titled "Microsoft Visual Studio 디버그 콘솔". The output text is as follows:

```
난생처음 C 프로그래밍  
D:\CProgramming\CSolution\x64\Release\CProject.exe(20540 프로세스)이  
(가) 0 코드로 인해 종료되었습니다.  
이 창을 닫으려면 아무 키나 누르세요.  
-
```

# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

7. [코드 2-1] 5행의 printf를 prnit로 잘못 입력한 경우,  
다음과 같은 창이 뜨면 '이 대화 상자를 다시 표시 안 함'에 체크 후  
<아니요>를 누름



# 01. 소스 파일과 헤더 파일이란?

## II. 소스 파일

8. 다시 프로그램을 실행하면 VS2022의 [출력] 창에 다음과 같은 메시지가 출력되고 빌드에 실패함

```
1>Main.obj : error LNK2001: print 외부 기호를 확인할 수 없습니다.  
1>fatal error LNK1120: 1개의 확인할 수 없는 외부 참조입니다.
```

# 01. 소스 파일과 헤더 파일이란?

## III. 헤더 파일

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

- `<stdio.h>` : 헤더 파일
- 컴퓨터가 소스 파일 Main.c를 읽을 때 `<stdio.h>`라는 헤더 파일을 참조하라는 의미
- 위 행을 삭제하면 아래와 같은 오류 문구가 출력됨

```
1>Main.obj : error LNK2001: printf 외부 기호를 확인할 수 없습니다.  
1>fatal error LNK1120: 1개의 확인할 수 없는 외부 참조입니다.
```

# 01. 소스 파일과 헤더 파일이란?

## 확인문제1

1. C 언어가 기록되는 파일의 주요 확장자 두 개는 무엇인가?
2. 소스 파일에서 헤더 파일을 참고하라는 의미로 사용되는 구문을 고르시오.  

① <code>#include</code>	② <code>stdio.h</code>
③ <code>printf</code>	④ <code>return 0;</code>
3. 다음 빈칸에 들어갈 단어를 채우시오.

소스 파일에서 들여쓰기나 칸을 띄워야 할 경우  이나  을 여러 번 사용할 수 있으며 문장의 끝에는  을 붙여야 한다.

# 01. 소스 파일과 헤더 파일이란?

4. 콘솔에 문자열을 출력하는 C 언어 명령인 `printf`를 알기 위하여 포함해야 하는 헤더 파일을 고르시오.

① `#include`

② `stdio.h`

③ `printf`

④ `return 0;`

5. 자신의 이름을 화면에 출력하기 위한 프로그램을 완성하시오.

```
01  _____ <stdio.h>
02
03  int main()
04  {
05      _____("자신의 이름");
06      return 0;
07  }
```



# 과제. 개쩌는 자기소개

---

## III. Main.c 파일에 자기소개를 작성하시오

- 이름, 취미, 특이사항, 게임티어... 등
- 본인을 잘 나타낼 수 있는 재미있고 참신한 자기소개
- 용기점수, 어그로점수 등 있음
- E-class 과제 란에 제출할것

02

C 코드의 기초

## 02. C 코드의 기초

### I. 코드를 구성하는 기본 요소

- 키워드(Keyword, 예약어)
  - 의미가 고정되어 다른 의미로 사용될 수 없는 단어

case	const	char	continue	default	do	double
else	enum	float	for	if	int	long
return	short	signed	sizeof	static	struct	switch
typedef	union	unsigned	void	while		

그림 2-2 자주 쓰는 키워드의 종류

## 02. C 코드의 기초

### I. 코드를 구성하는 기본 요소

- 식별자(Identifier)

- 프로그램을 구성하는 각종 요소(변수, 함수 등)를 식별하기 위하여 프로그래머가 직접 지은 이름

- 식별자를 만드는 규칙

1. 키워드(예약어)는 식별자가 될 수 없음
2. 첫 문자로 숫자가 올 수 없음
3. 대소문자를 구분함

## 02. C 코드의 기초

### I. 코드를 구성하는 기본 요소

- 리터럴(Literal)

- 그 자체로서 값을 나타내는 것

- 리터럴의 종류

- 숫자 리터럴 : 숫자를 의미

- 문자 리터럴 : 작은따옴표('')로 이루어진 한 개의 문자, 숫자를 의미

- 문자열 리터럴 : 큰따옴표("")로 이루어진 문자열을 의미

## 02. C 코드의 기초

### I. 코드를 구성하는 기본 요소

- 연산자(Operator)

- 사칙연산 기호인 +, -, \*, / 등이 있음
- 수학의 연산자와 차이가 있음
- 기호 =의 경우 '대입'을 나타냄

- 문장 부호(Punctuator)

- 작은따옴표(''), 큰따옴표(""), 중괄호({}), 세미콜론(;) 등

## 02. C 코드의 기초

### I. 코드를 구성하는 기본 요소

- 토큰 분석하기

```
int a = b + 1;
```

- [int, a, =, b, +, 1, ;] 총 7개의 토큰으로 분리됨
- int : 키워드
- a, b : 식별자
- =, + : 연산자
- 1 : 리터럴
- ; : 문장부호

## 02. C 코드의 기초

### II. 정수 리터럴과 진법

- 기본적으로 10진법을 가정
- 숫자 앞에 진법을 나타내는 접두사를 붙여  
다른 진법을 숫자를 나타낼 수 있음

표 2-1 진법과 접두사

진법	접두사	표기	값(10진법 기준)
10진법	없음	11	$10 * 1 + 1 * 1 = 11$
2진법	0b	0b11	$2 * 1 + 1 * 1 = 3$
8진법	0	011	$8 * 1 + 1 * 1 = 9$
16진법	0x	0x11	$16 * 1 + 1 * 1 = 17$



## 02. C 코드의 기초

### II. 정수 리터럴과 진법

- 2진법

- 접두사 0b가 있으면 2진법 리터럴로 해석됨
- 컴퓨터 메모리의 최소 단위인 비트(0 혹은 1) 하나가 2진법 숫자 하나를 그대로 나타낼 수 있음
- 메모리의 상태를 설정할 때 많이 사용됨

- 8진법

- 접두사 0이 있으면 8진법 리터럴로 해석됨
- 8진법 숫자 하나는 3비트( $2^3$ )로 표현됨
- 컴퓨터는 2의 승수를 주로 사용하여 8진법 리터럴은 잘 사용되지 않음

## 02. C 코드의 기초

### II. 정수 리터럴과 진법

- 16진법
  - 접두사 0x가 있으면 2진법 리터럴로 해석됨
  - 2진법과 대응할 수 있어 10진법 다음으로 많이 사용
- 2진법을 16진법으로 바꾸는 법
  - 2진법의 숫자 4개를 묶어서 16진법 숫자 하나로 변환

## 02. C 코드의 기초

### II. 정수 리터럴과 진법

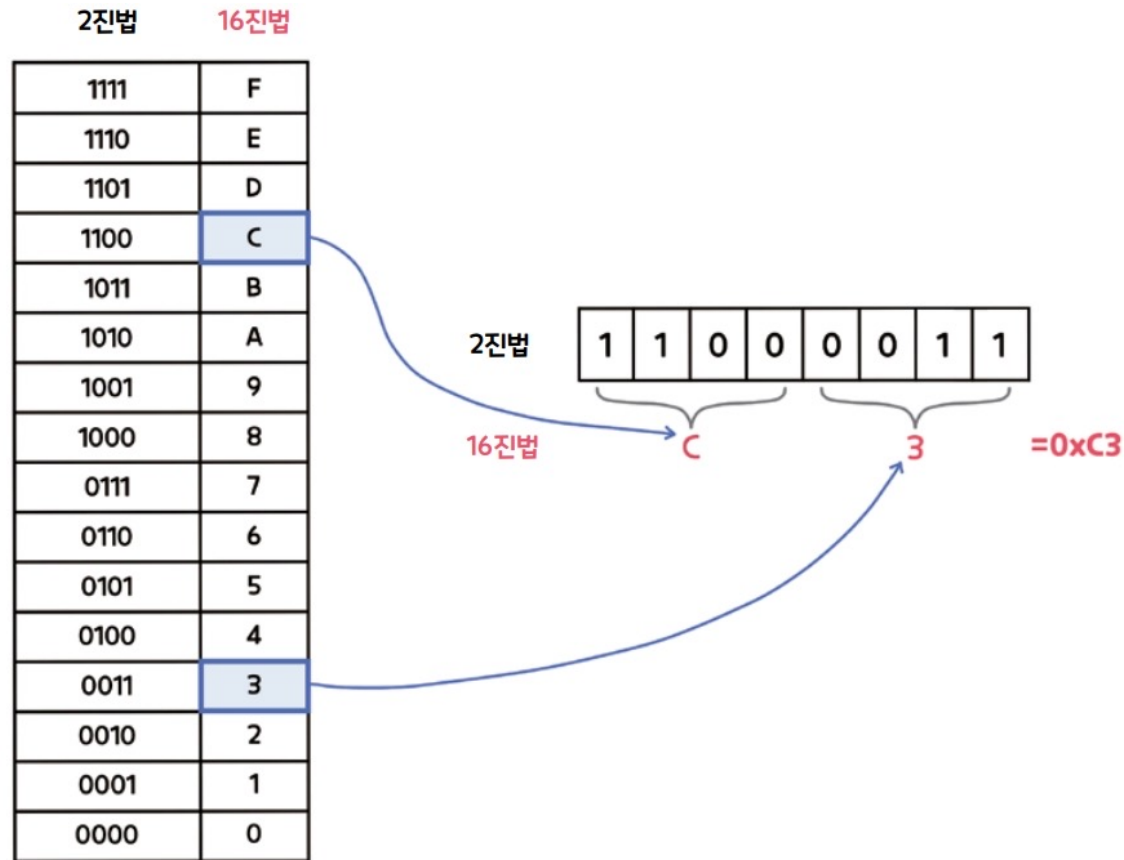


그림 2-3 2진법에서 16진법으로 변환하는 방법

## 02. C 코드의 기초

### III. 주석(Comments)

- 주석의 개념
  - 코드의 의미를 설명하는 등의 용도로 코드에 쓰인 것
- 주석의 형식
  - `//` : 한 줄 주석, 해당 줄의 모든 문자들을 주석 처리
  - `/**/` : 여러 줄 주석, `/*`와 `*/` 사이의 모든 문자들을 주석 처리

## 02. C 코드의 기초

### III. 주석(Comments)

- 주석의 활용

#### [코드 2-2] 주석

```
01  #include <stdio.h>
02
03  int main() // 한 줄 주석입니다
04  {
05      /* 여러 줄 주석으로 이런 식으로 코드를 무효 처리할 수 있습니다.
06      printf("난생처음 c 프로그래밍");
07      */
08      return 0;
09  }
```

## 02. C 코드의 기초

### 확인문제2

1. 식별자로 올바르지 못한 것을 고르시오.

① \_name

② nAme

③ name2

④ 2name

2. 10진수 123을 2진법과 16진법으로 나타내시오.

(1) 2진법 : \_\_\_\_\_

(2) 16진법 : \_\_\_\_\_

3. 다음 문장을 토큰으로 분리하고 각 토큰의 종류를 설명하시오.

```
str = "int a = 0x1;";    // a는 1;
```

03

C 프로그램의 기초

## 03. C 프로그램의 기초

### I. 함수(Function)

- 함수의 개념

- 특정 기능을 수행하는 코드의 묶음
- x 값이 정해지면 연산 결과에 따라 y 값이 자연스럽게 정해짐

$y = x + 1$



[코드 2-3]  $y = x + 1$

```
01  int y(int x)
02  {
03      return x + 1;
04  }
```



## 03. C 프로그램의 기초

### I. 함수(Function)

- 함수의 개념

- 함수 본체에 함수값을 반환하는 것 외에도 추가적인 처리가 가능함
- 따라서 함수 본체의 시작과 끝을 중괄호({})를 사용하여 명확히 함

#### [코드 2-4] 함수 본체의 추가적인 처리

```
01  int y(int x)
02  {
03      printf("변수: %d", x);
04      return x + 1;
05  }
```

# 03. C 프로그램의 기초

## I. 함수(Function)

- 함수의 개념

`sum(a, b) = a + b`



### [코드 2-5] 매개변수 2개인 함수

```
01  int sum(int a, int b)
02  {
03      int c = a + b;
04      return c;
05  }
```

## 03. C 프로그램의 기초

### I. 함수(Function)

- 함수의 개념
  - 인자를 전달하여 함수를 호출할 수 있음

```
int result = sum(1, 2);
```



인자

# 03. C 프로그램의 기초

## II. main 함수

- C 프로그램의 시작 함수(Entry Function)의 이름
- main 함수의 두 가지 형식
  - <표준 형식 1> : 매개 변수가 없음
  - <표준 형식 2> : 매개 변수가 2개

<표준 형식 1>

```
int main()  
{  
    ...  
}
```

<표준 형식 2>

```
int main(int argc, char** argv)  
{  
    ...  
}
```

# 03. C 프로그램의 기초

## II. main 함수

[코드 2-6] 한 프로그램에 main 함수가 두 개인 경우

```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("난생처음");
06      return 0;
07  }
08
09  int main(int argc, char** argv)
10  {
11      printf("C 프로그래밍");
12      return 0;
13  }
```

1>error C2084: 'int main()' 함수에 이미 본문이 있습니다.  
1>'main'의 이전 정의를 참조하십시오.

# 03. C 프로그램의 기초

## II. main 함수

### 하나 더 알기

### main에서 return을 생략하는 경우

- [코드 2-7]은 [코드 2-1]의 main 함수 본체 끝에 return 0; 구문을 삭제한 것
- Main 함수 본체에서 명시적으로 return이 보이지 않을 경우, 컴파일러가 기본으로 0이 반환되는 코드를 추가해줌
- 따라서 main에서 return을 생략할 수 있음

### [코드 2-7] main에서 return을 생략하는 경우

```
01 #include <stdio.h>
02
03 int main()
04 {
05     printf("난생처음 c 프로그래밍");
06 }
```

## 03. C 프로그램의 기초

### LAB 2-1

### 두 수를 곱하는 함수

두 개의 정수를 곱하여 반환하는 Multiply 함수를 작성해봅시다.

- 1 함수의 이름은 `multiply` 그대로 사용한다.
- 2 변수 `i`, `j`가 모두 정수이므로 코드로는 `int i`, `int j`로 표시한다.
- 3 정수끼리의 곱은 정수이므로 함수값도 정수이므로 `int multiply(int i, int j)`이다.
- 4 `return`을 이용하여 `i * j`를 반환한다.

## 03. C 프로그램의 기초

LAB 2-1

정답

```
01  int multiplay(int I, int j)
02  {
03      return i*j;
04  }
```



04

표준 입출력이란?

## 04. 표준 입출력이란?

### I. 시스템 콘솔

- 표준 입출력

- 가장 기본적인 장치를 사용하는 입출력
- 키보드로 입력을 받고 모니터로 출력하는 것

- 시스템 콘솔의 개념

- [검색]에서 cmd를 입력 후 나타나는 [명령 프롬프트]가 텍스트 기반의 입출력 수행 시스템 콘솔

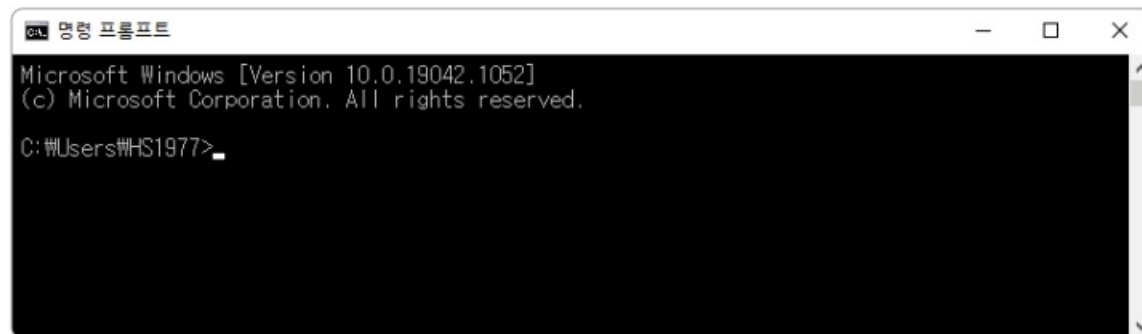


그림 2-4 명령 프롬프트

## 04. 표준 입출력이란?

### II. 표준 출력 함수 printf

- printf 함수의 개념
  - 콘솔에 문자열을 출력하는 함수
- printf 함수의 형태

```
int printf(const char* format, ...);
```

## 04. 표준 입출력이란?

### II. 표준 출력 함수 printf

#### [코드 2-8] printf

```
01 #include <stdio.h>
02
03 int main()
04 {
05     printf("%d 난생처음 %s", 2022, "C 프로그래밍");
06 }
```

2022 난생처음 C 프로그래밍

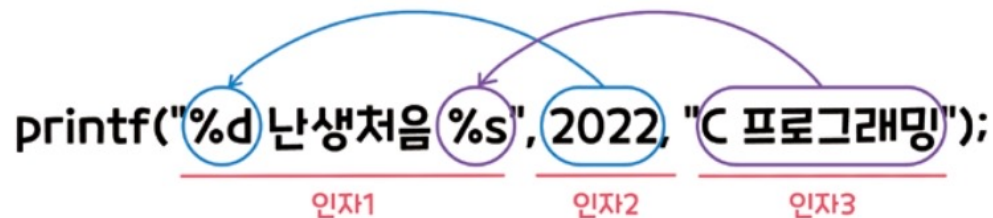


그림 2-5 printf에 전달되는 인자

# 04. 표준 입출력이란?

## II. 표준 출력 함수 printf

- 출력 제어 문자(Escape Sequence)
  - 콘솔에 문자열을 출력하는 함수

### [코드 2-9] 두 줄 출력을 위한 시도

```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("%d 난생처음 %s");
06      printf("c 프로그래밍");
07  }
```

난생처음C 프로그래밍

# 04. 표준 입출력이란?

## II. 표준 출력 함수 printf

- 출력 제어 문자(Escape Sequence)
  - printf를 한 번만 호출해도 두 줄에 걸쳐서 출력할 수 있음

### [코드 2-10] 출력 제어 문자

```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("난생처음\r\nC 프로그래밍");
06  }
```

난생처음  
C프로그래밍

## 04. 표준 입출력이란?

### II. 표준 출력 함수 printf

- 출력 제어 문자의 종류

표 2-2 출력 제어 문자(Escape Sequence)의 종류

출력 제어 문자	설명
\0	널(NULL) 문자, 문자열의 끝을 표시함
\b	백스페이스(Backspace), 커서 위치에서 한 칸 뒤로 이동
\t	탭(Tab) 출력
\n	줄 바꿈(Line Feed), 한 줄 넘어감
\r	캐리지 리턴(Carriage Return), 커서를 현재 라인 시작 위치로 옮김
\"	큰따옴표 출력
\'	작은따옴표 출력
\\	백슬래시(Backslash) 출력

## 04. 표준 입출력이란?

### III. 표준 입력 함수 scanf

- scanf 함수의 형태
  - 콘솔로부터 키보드 입력을 받는 함수
  - printf와 형태가 같음

```
int scanf(const char* format, ...);
```



## 04. 표준 입출력이란?

### III. 표준 입력 함수 scanf

#### [코드 2-11] scanf 함수

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06      char name[16] = { 0 };
07      int age;
08
09      printf("이름이 뭐예요?\r\n");
10      scanf("%s", name);
11      printf("나이가 어떻게 되시죠?\r\n");
12      scanf("%d", &age);
13
14      printf("이름: %s\r\n나이: %d", name, age);
15  }
```

이름이 뭐예요?

홍길동

나이가 어떻게 되시죠?

20

이름: 홍길동

나이: 20

## 04. 표준 입출력이란?

### III. 표준 입력 함수 scanf

#### [코드 2-12] scanf 한 번 호출

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06      char name[16] = { 0 };
07      int age;
08
09      printf("이름과 나이를 공백으로 구분하여 입력하세요.\r\n");
10      scanf("%s%d", name, &age);
11
12      printf("이름: %s\r\n나이: %d", name, age);
13  }
```

이름과 나이를 공백으로 구분하여 입력하세요.  
홍길동 20  
이름: 홍길동  
나이: 20

## 04. 표준 입출력이란?

### 확인문제3

1. 다음 빈칸에 들어갈 단어를 채우시오.

시스템 콘솔에서 일어나는 입출력을  이라고 합니다. 윈도우의 경우 명령 프롬프트가 시스템 콘솔입니다.

2. 다음과 같이 출력되는 printf문으로 올바른 것을 고르시오(단, 정수인 변수 a, b에는 2, 3이 저장되어 있다.).

2 \* 3 = 6

- ① `printf("%s * %d = %d", a, b, a * b);`
  - ② `printf("%d * %d = %d", a, b, a * b);`
  - ③ `printf("%d * %s = %d", a, b, a * b);`
  - ④ `printf("%d * %d = %s", a, b, a * b);`
3. 공백으로 구분되는 입력으로 태어난 연도, 월, 일, 성별, 이름을 변수 year, month, day, gender, name으로 받도록 scanf 함수를 작성하시오. 단, 성별과 이름은 문자열이다.

## 04. 실수형

### LAB 2-2

### 입력받은 두 수를 곱하고 출력하기

두 수를 입력받아 곱한 값을 출력하는 프로그램을 작성해봅시다.

두 정수를 공백으로 구분하여 입력하세요.

3 7

두 수의 곱은 21입니다.

- 1 scanf를 사용하기 위한 구문을 반드시 작성한다.
- 2 입력받은 두 정수를 저장할 변수 a, b를 마련한다.
- 3 scanf에 서식 문자열로 "%d%d"를 사용하여 공백으로 두 수를 입력받고, 각각 변수 a와 b에 저장한다. 이때 &a, &b를 넘긴다.
- 4 변수 c를 마련하여 두 수의 곱을 구한 뒤에 변수 c를 출력한다.

## 04. 실수형

### LAB 2-2

### 정답

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06      int a;
07      int b;
08      printf("두 정수를 공백으로 구분하여 입력하세요.\r\n");
09
10      scanf("%d%d", &a, &b);
11
12      int c = a * b;
13      printf("두 수의 곱은 %d입니다.", c);
14  }
```

05

프로그램 생성 및 실행

## 05. 프로그램 생성 및 실행

---

### I. 간단한 메아리 프로그램

- 메아리(Echo) 프로그램
  - 입력(요청)한 것을 그대로 출력(응답)하는 프로그램
  - 네트워크 프로그래밍 시 클라이언트와 서버 사이의 입출력이 잘 되는지 간단히 확인할 때 사용

# 05. 프로그램 생성 및 실행

## I. 간단한 메아리 프로그램

[코드 2-13] 메아리 프로그램

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main( )
05  {
06      char input[128] = { 0 };
07
08      printf("입력> ");
09      scanf("%[^\n]s", input);
10
11      printf("출력> %s", input);
12  }
```

입력> 난생처음 C 프로그래밍

출력> 난생처음 C 프로그래밍



# 05. 프로그램 생성 및 실행

## II. 생성된 프로그램 위치 확인하기

- 프로젝트 속성 페이지의 설정 값에 따라 프로젝트를 빌드하여 생성된 프로그램의 위치가 결정됨

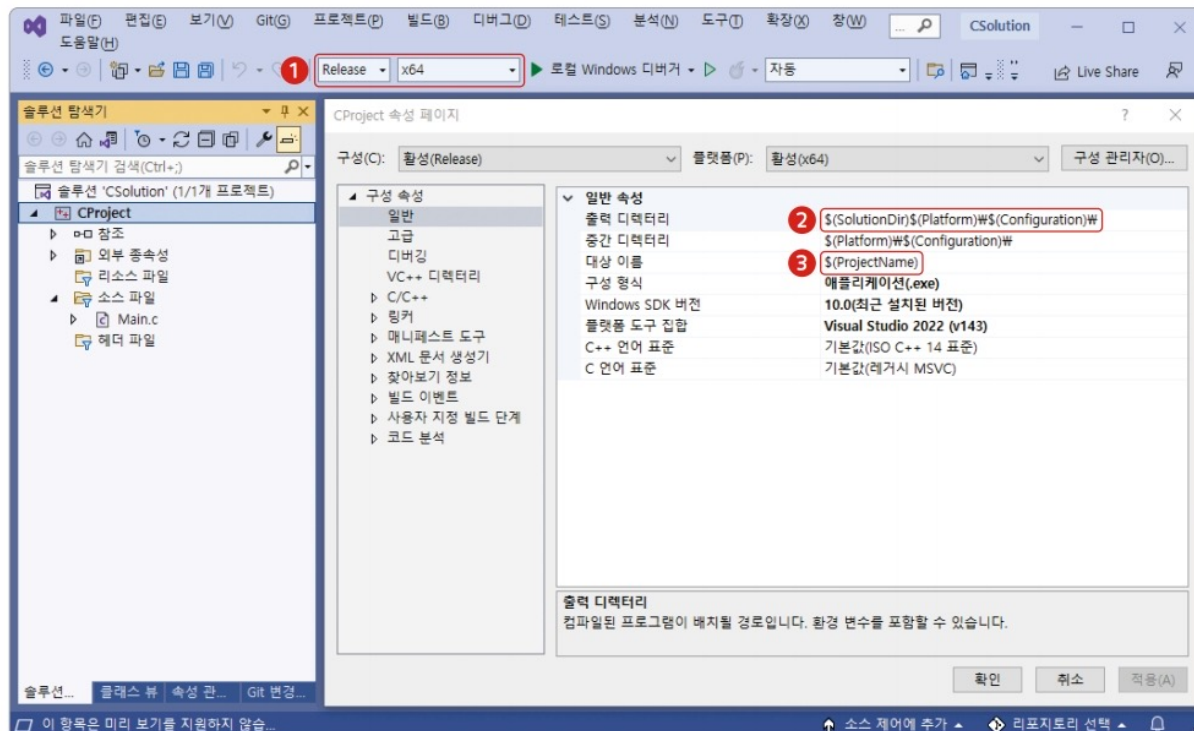


그림 2-7 비주얼 스튜디오 프로젝트 속성 페이지

# 05. 프로그램 생성 및 실행

## II. 생성된 프로그램 위치 확인하기

- 프로젝트 실행 디렉터리

D:\CProgramming\CSolution\CProject

- 솔루션 디렉터리
  - 솔루션 파일이 있는 곳
  - 경로 : D:\CProgramming\CSolution
  - 솔루션 파일 : CSolution.sln

# 05. 프로그램 생성 및 실행

---

## II. 생성된 프로그램 위치 확인하기

- 프로젝트 디렉터리
  - 프로젝트 파일이 있는 곳
  - 경로 : D:\WCProgramming\WCSolution\WCProject
  - 프로젝트 파일 : CProject.vcxproj

# 05. 프로그램 생성 및 실행

## II. 생성된 프로그램 위치 확인하기

- 프로젝트 디렉터리

- [그림 2-7]의 ①, ②, ③ 해석

```
① Release, x64
② $(SolutionDir)$(Platform)\$(Configuration)\
③ $(ProjectName)
```

- \$(SolutionDir) : 솔루션 디렉터리 즉 :WCProgrammingWCSolutionW를 의미함
- \$(Platform) : ①의 64비트 시스템을 의미함
- \$(Configuration) : ①의 Release를 나타냄
- \$(ProjectName) : 프로젝트 이름 즉 CProject를 의미함
- 프로그램의 경로 : ②와 ③을 합친 경로

# 05. 프로그램 생성 및 실행

## II. 빌드 및 프로그램 실행하기

- VS2022 메뉴에서 실행하기

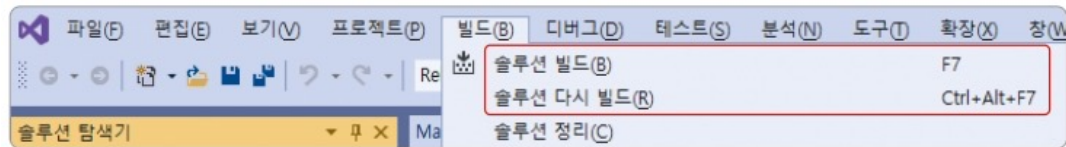


그림 2-8 솔루션 빌드 및 솔루션 다시 빌드

# 05. 프로그램 생성 및 실행

## II. 빌드 및 프로그램 실행하기

- 탐색기를 이용하여 실행하기

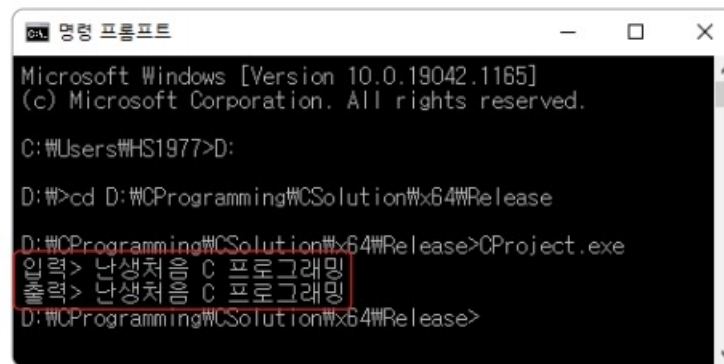
- 1) 탐색기에서 프로그램이 생성된 폴더로 이동  
(D:\WCProgramming\WCSolution\wx64\Release)
- 2) CProject.exe 파일 더블클릭하여 실행
- 3) 프로그램 실행 후 입력하여 콘솔창이 닫혀 출력 확인이 불가능한 경우,  
윈도우 [검색]에서 cmd를 입력 후 [명령 프롬프트]를 선택하여 직접 콘솔 열기

# 05. 프로그램 생성 및 실행

## II. 빌드 및 프로그램 실행하기

- 탐색기를 이용하여 실행하기

- 4) 프로그램 파일이 있는 디렉터리로 이동
- 5) 드라이브가 다른 경우 D:를 입력하여 드라이브 변경
- 6) cd(change directory) 바로 뒤에 이동할 경로를 입력
- 7) CProject.exe 입력하면 프로그램이 실행됨



```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HS1977>D:
D:\>cd D:\CProgramming\CSolution\x64\Release
D:\CProgramming\CSolution\x64\Release>CProject.exe
입력> 난생처음 C 프로그래밍
출력> 난생처음 C 프로그래밍
D:\CProgramming\CSolution\x64\Release>
```

그림 2-9 콘솔에서 프로그램 실행

## 05. 프로그램 생성 및 실행

### 확인문제4

윈도우의 시스템 콘솔인 명령 프롬프트에서 C:\Windows\System32\drivers\etc로 이동하기 위한 cd 명령어는 무엇인가?



# [실전예제]

두 수를 활용한 세 가지 연산

# [실전예제] 두 수를 활용한 세 가지 연산

## [문제]

두 수를 입력받아 곱하기(+), 더하기(+), 빼기(-)를 출력하는 프로그램을 작성해봅시다. 이때 곱하기, 더하기, 빼기는 각각의 함수를 따로 구현하여 호출하여 사용합니다.

### 실행 결과

두 정수를 공백으로 구분하여 입력하세요.

7 3

곱: 21

합: 10

차: 4

# [실전예제] 두 수를 활용한 세 가지 연산

## [해결]

1. 두 정수의 곱, 합, 차를 구하는 매개변수 두 개인 함수 `multiply`, `add`, `subtract`를 작성한다.
2. 정수 변수 `a`, `b`를 마련하고 서식 문자열 `"%d%d"`로 `scanf`를 호출하여 `&a`, `&b`를 인자로 전달한다.
3. 변수 `c1`, `c2`, `c3`를 마련하고 각각 세 함수를 호출한 결과를 저장한다.
4. 서식 문자열 안에 출력 제어 문자열 `\r\n`을 사용하여 단 한 번의 `printf`를 호출하여 세 줄에 걸쳐서 `c1`, `c2`, `c3`가 출력되도록 한다.

# [실전예제] 두 수를 활용한 세 가지 연산

## [해결]

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int multiply(int i, int j)
05  {
06      return i * j;
07  }
08
09  int add(int i, int j)
10  {
11      return i + j;
12  }
13
14  int subtract(int i, int j)
15  {
16      return i - j;
17  }
```

# [실전예제] 두 수를 활용한 세 가지 연산

## [해결]

```
14  int main( )
15  {
16      int a;
17      int b;
18      printf("두 정수를 공백으로 구분하여 입력하세요.\r\n");
19      scanf("%d%d", &a, &b);
20
21      int c1 = multiply(a, b);
22      int c2 = add(a, b);
23      int c3 = subtract(a, b);
24      printf("곱: %d\r\n합: %d\r\n차: %d", c1, c2, c3);
25  }
```

# Thank you!