

Chapter 02

일단 짜보는 그럴듯한

C 프로그램



목차

1. 프로그램 작성 순서 복습
2. scanf() 함수 맛보기

01

프로그램 작성 순서 복습

1. 프로그램 작성 순서 복습



그림 2-1 프로그램 작성 순서

1. 두 번째 프로젝트 만들기

- 프로젝트 이름은 'Second'

01 Visual Studio를 실행

1. 프로그램 작성 순서 복습

02 [시작 화면] 오른쪽 아래 [새 프로젝트 만들기]를 클릭 →

[모든 언어]를 [C++] 선택 → [Windows 데스크톱 마법사] 선택 → <다음> 클릭 →
프로젝트 이름을 'Second'로 입력 → 위치는 'C:\WCookC'를 입력하거나 선택 →
'솔루션 및 프로젝트를 같은 디렉터리에 배치'에 체크 → <만들기> 클릭

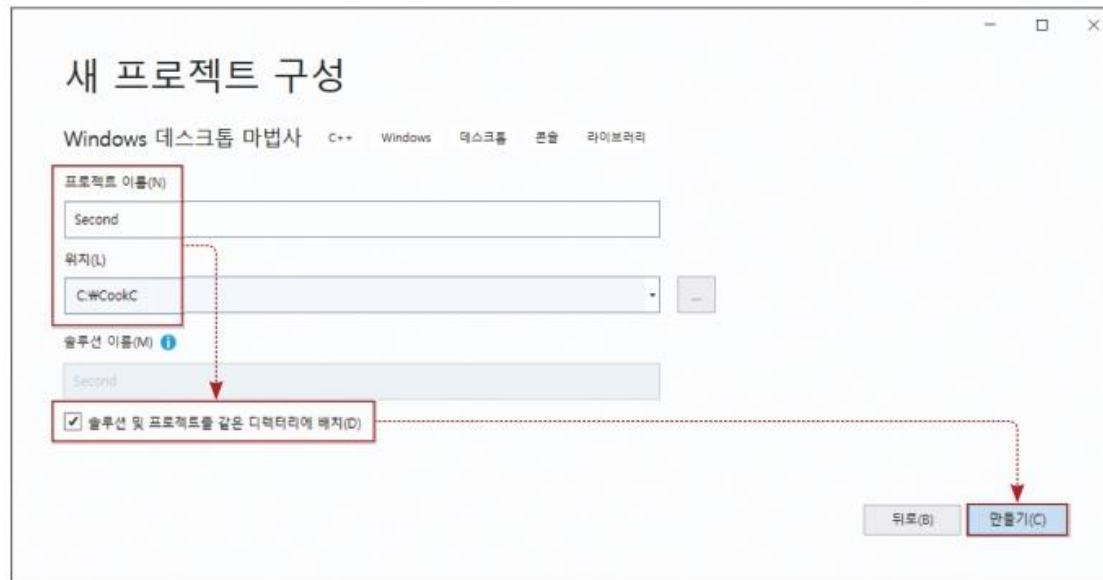


그림 2-2 새 프로젝트인 Second 생성

1. 프로그램 작성 순서 복습

03 [Windows 데스크톱 프로젝트] 창에서 애플리케이션 종류는 '콘솔 애플리케이션 (.exe)'을 선택하며, 추가 옵션으로 '빈 프로젝트'에 체크하고 <확인>을 클릭

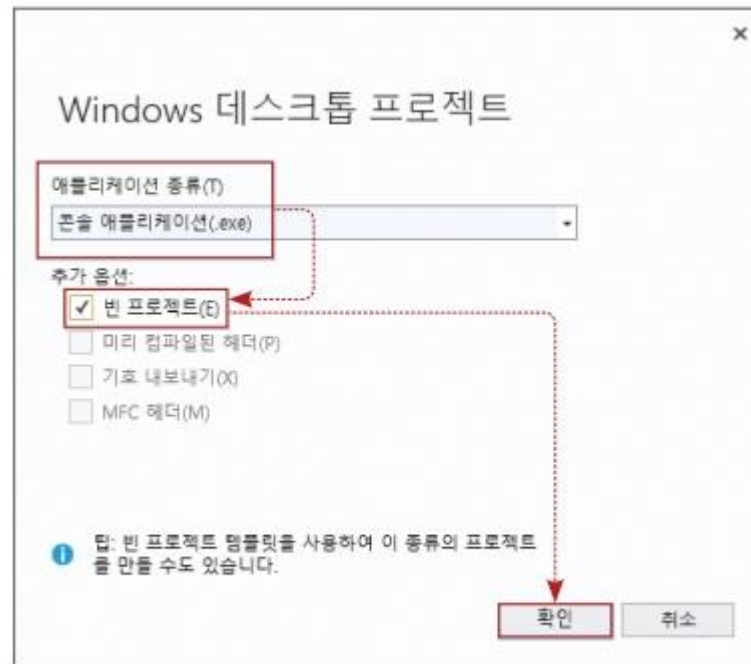


그림 2-3 응용 프로그램 설정

1. 프로그램 작성 순서 복습

04 다음과 같이 빈 프로젝트가 완성, 이 프로젝트의 이름은 'Second'

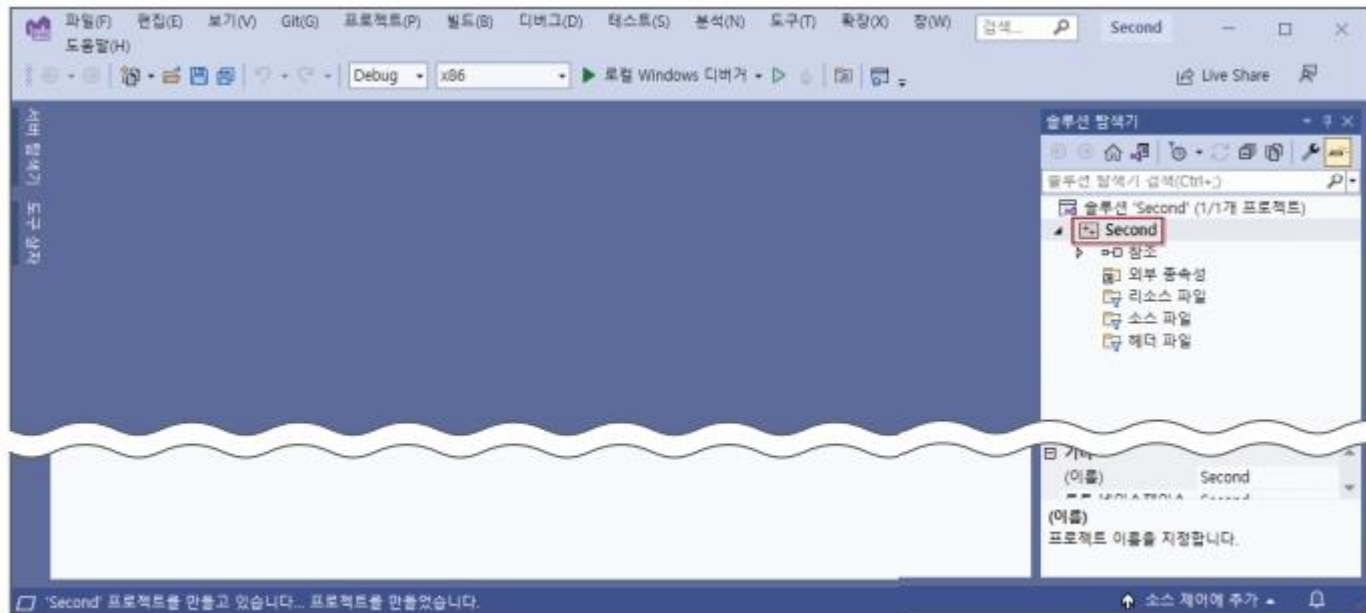


그림 2-4 생성된 Second 프로젝트

05 메뉴의 [파일] - [끝내기]를 선택하여 Visual Studio를 종료

SELF STUDY

'Test2', 'Test3'이라는 이름의 프로젝트 2개를 빠른 속도로 만들어보자.

1. 프로그램 작성 순서 복습

2. 프로그램 코딩

01 Visual Studio를 실행

02 Visual Studio의 오른쪽 [시작] 부분의 [프로젝트 또는 솔루션 열기]를 선택한 후 앞서 작업했던 'C:\W\CookCW\Second' 폴더의 'Second.sln' 파일을 선택

03 오른쪽 [솔루션 탐색기]의 프로젝트 이름(현재 Second) 아래 '소스 파일' 폴더에서 마우스 오른쪽 버튼을 클릭하여 [추가] - [새 항목]을 선택

04 [새 항목 추가] 창의 왼쪽에서 [Visual C++]를 선택한 후, 오른쪽의 'C++ 파일(.cpp)'을 선택한 상태에서 이름에 'Second.c'를 입력하고 <추가>를 클릭

05 100과 50의 더하기, 빼기, 곱하기, 나누기를 수행하는 프로그램을 [기본 2-1]과 같이 코딩

1. 프로그램 작성 순서 복습

2. 프로그램 코딩

기본 2-1 두 번째로 만드는 C 프로그램

2-1.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int a, b;
06     int result;
07
08     a=100;
09     b=50;
10
11     result = a + b;
12     printf(" %d + %d = %d \n", a, b, result);
13
14     result = a - b;
15     printf(" %d - %d = %d \n", a, b, result);
16
```

----- 계산할 두 수를 저장할 변수 a, b와
결과를 넣을 변수 result를 선언한다.

----- a에 100을, b에 50을 넣는다.

----- a와 b를 더한 결과를 result에 넣는다.

----- 모니터에 a, b, result를 출력한다.

----- a와 b를 뺀 결과를 result에 넣는다.

----- 모니터에 a, b, result를 출력한다.

1. 프로그램 작성 순서 복습

2. 프로그램 코딩

```
17  result = a * b;      —— a와 b를 곱한 결과를 result에 넣는다.
18  printf(" %d * %d = %d \n", a, b, result); —— 모니터에 a, b, result를 출력한다.
19
20  result = a / b;      —— a와 b를 나눈 결과를 result에 넣는다.
21  printf(" %d / %d = %d \n", a, b, result); —— 모니터에 a, b, result를 출력한다.
22 }
```

1. 프로그램 작성 순서 복습

2. 프로그램 코딩

06 틀린 글자가 없는지 확인한 후 메뉴의 [파일]-[모두 저장]을 선택하여 저장

변수는 '값을 저장하는 그릇(또는 방)'

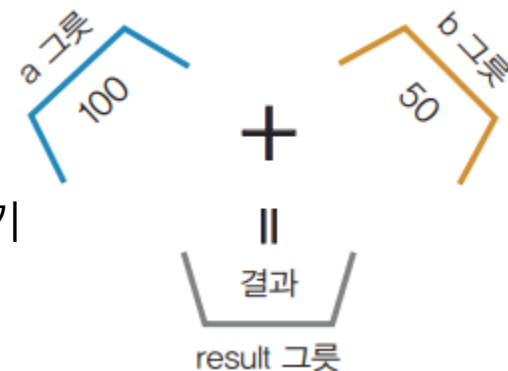
- 5행과 6행 : 변수(그릇)가 3개 등장



- 8행과 9행 : a 그릇에는 100을 넣고 b 그릇에는 50을 넣기



- 11행 : a 그릇 값과 b 그릇 값을 더한 결과를 result 그릇에 넣기



1. 프로그램 작성 순서 복습

2. 프로그램 코딩

06 12행의 내용 중 printf()는 괄호 안의 내용을 모니터에 출력하라는 의미

100, 50, 150은 모두 정수이므로 그에 대응하는 %d를 사용

맨 뒤에 있는 \n은 실제로 출력되지 않는 기호로서 다음 줄로 넘기라는 의미

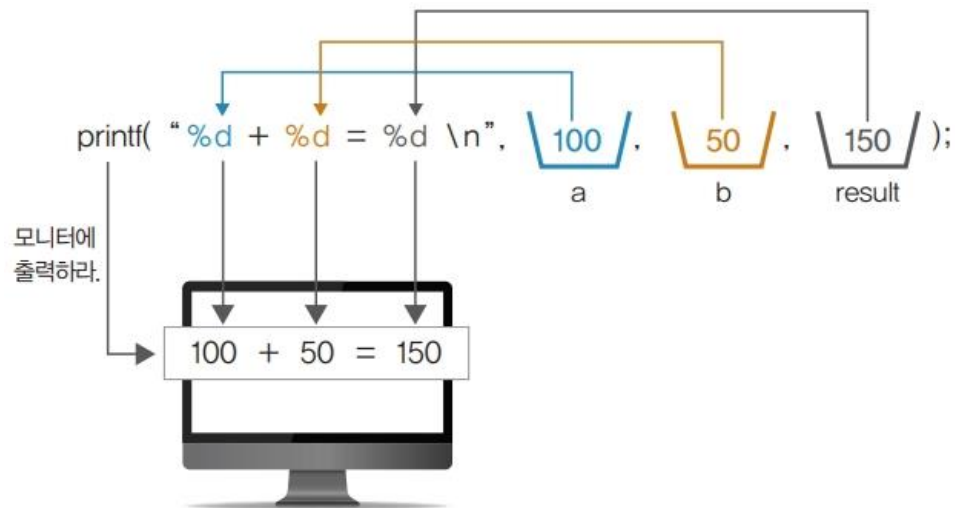


그림 2-5 printf() 함수의 의미

1. 프로그램 작성 순서 복습

3. 빌드(컴파일/링크)

01 메뉴의 [빌드] - [솔루션 빌드]를 선택해서 프로젝트를 빌드

02 특별히 문제가 없다면 다음 그림과 같이 '성공 1, 실패 0, 최신 0, 생략 0'이 출력

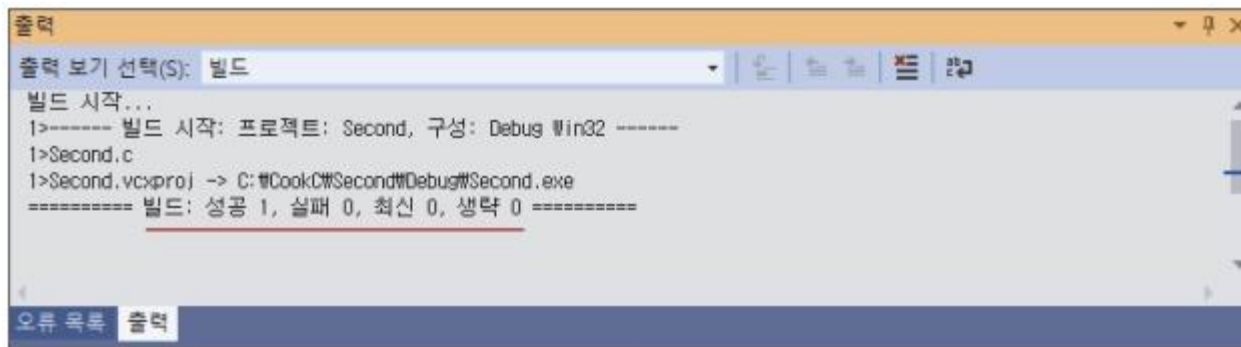


그림 2-6 프로젝트의 빌드 성공

03 만약, 실패가 나오면 소스에서 틀린 부분이 있다는 뜻이므로 소스에서 틀린 부분을 찾아 수정한 후 다시 빌드

여기서 잠깐 변수와 그릇의 차이

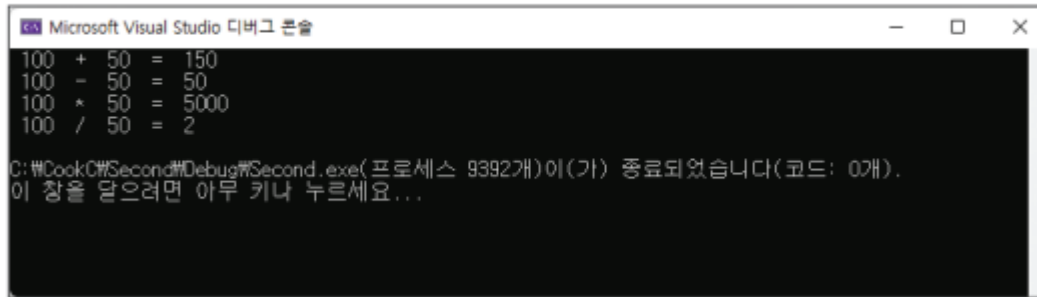
그릇과 변수의 차이점은, a 그릇과 b 그릇의 값인 100과 50은 없어지지 않고 그대로 있지만 실제 물이 담긴 그릇이라면 물이 없어진다는 것

1. 프로그램 작성 순서 복습

4. 실행

01 [Ctrl] + [F5] 를 눌러서 실행

실행 결과▼



```
Microsoft Visual Studio 디버그 콘솔
100 + 50 = 150
100 - 50 = 50
100 * 50 = 5000
100 / 50 = 2
C:\북Cook\CHSecond\Debug\Second.exe( 프로세스 9392개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

02 결과를 확인한 후 아무 키나 누르면 결과 창이 닫힘

SELF STUDY

[기본 2-1]을 수정하여 1234와 456의 더하기, 빼기, 곱하기, 나누기 결과를 확인해보자.

힌트_ 변수 a와 b에 대입하는 값을 변경하면 된다.

02

scanf() 함수 맛보기

2. scanf() 함수 맛보기

1. 값을 입력받는 scanf() 함수

여기서 잠깐 scanf()와 scanf_s() 함수

- C에서는 전통적으로 값을 입력받기 위해 scanf() 함수를 사용
- 마이크로소프트에서는 scanf() 함수 대신 scanf_s() 함수의 사용을 적극 권장
- scanf()와 scanf_s() 함수의 용도는 동일하지만 scanf() 함수가 사용되는 소스의 1행에는 `#define _CRT_SECURE_NO_WARNINGS`라는 코드를 추가해야 함
- scanf_s() 함수의 단점은 Visual C++에서만 사용할 수 있으며 다른 C 컴파일러는 인식 하지 못한다는 것
- scanf_s()와 같이 기존의 함수 이름에 '_s'가 붙은 새로운 함수는 보안이 한층 강화된 개선된 함수

a에 100을 넣는다. ⇨ a에 입력할 값을 키보드로 입력받는다.

b에 50을 넣는다. ⇨ b에 입력할 값을 키보드로 입력받는다.



```
Microsoft Visual Studio 디버그 콘솔
첫번째 계산할 값을 입력하세요 ==> 100
두번째 계산할 값을 입력하세요 ==> 50
100 + 50 = 150
100 - 50 = 50
100 * 50 = 5000
100 / 50 = 2
C:\Cook\CS\Second\Debug\Second.exe (프로세스 9580개)이(가) 종료되었습니다(코드: 0x0).
이 창을 닫으려면 아무 키나 누르세요...
```

그림 2-7 사용자가 직접 값을 입력하는 프로그램

2. scanf() 함수 맛보기

01 값을 입력받는 scanf_s() 함수를 사용

[기본 2-1]의 8행과 9행을 아래 표시된 부분과 같이 수정

응용 2-2 소스 수정하기(키보드로 값을 입력받음)

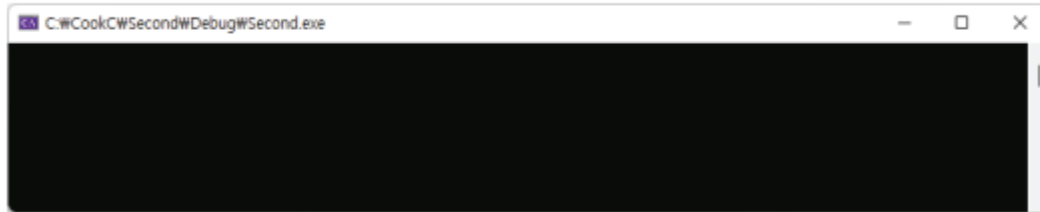
2-2.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     int result;
07
08     scanf("%d", &a);      ----- 키보드로 a에 들어갈 값을 입력받는다.
09     scanf("%d", &b);      ----- 키보드로 b에 들어갈 값을 입력받는다.
10
11     result = a + b;
12     printf(" %d + %d = %d \n", a, b, result);
13
14     ~~~ 이하는 [기본 2-1]의 14~21행과 동일함 ~~~
15     :
22 }
```

2. scanf() 함수 맛보기

02 [Ctrl]+[F5]를 눌러서 빌드와 실행을 동시에 진행

실행 결과 ▼



03 숫자를 하나 입력하고 [enter]를 누름. 다시 숫자 하나를 입력하고 [enter]

실행 결과 ▼



2. scanf() 함수 맛보기

04 프로그램을 좀 더 편하게 사용하기 위해 다음과 같이 수정

응용 2-3 소스 수정하기(도움말 출력)

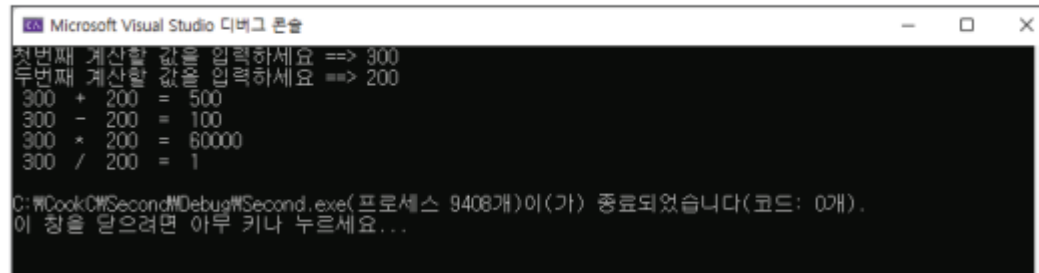
2-3.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     int result;
07
08     printf("첫 번째 계산할 값을 입력하세요 ==> "); ----- 도움말을 화면에 출력한다.
09     __1__ ("%d", &a);
10
11     printf("두 번째 계산할 값을 입력하세요 ==> "); ----- 도움말을 화면에 출력한다.
12     __2__ ("%d", &b);
13
14     result = a + b ;
15     __3__ (" %d + %d = %d \n",a ,b ,result);
16
17     ~~~ 이하는 [기본 2-1]의 14~21행과 동일함 ~~~
18     :
25 }
```

2. scanf() 함수 맛보기

05 다시 [Ctrl]+[F5]를 눌러서 빌드와 실행을 동시에 진행

실행 결과 ▼



```
Microsoft Visual Studio 디버그 콘솔
첫번째 계산할 값을 입력하세요 ==> 300
두번째 계산할 값을 입력하세요 ==> 200
300 + 200 = 500
300 - 200 = 100
300 * 200 = 60000
300 / 200 = 1
C:\Wook\CS\Second\Debug\Second.exe(프로세스 9408개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

SELF STUDY

값을 3개 입력받아 덧셈과 곱셈을 수행하도록 [응용 2-3]을 수정해보자.

힌트_ 변수 a, b와 동일하게 새로운 변수 c를 사용한다.

여기서 잠깐 한번에 하는 저장, 빌드, 실행

[Ctrl]+[F5]는 변경된 코드의 저장, 빌드, 실행을 동시 진행하는 단축키

*

예제 모음

[예제모음 01] 숫자 4개를 더하는 프로그램

예제 설명 숫자 4개를 입력받아 그 합을 구하는 프로그램이다.

실행 결과

첫 번째 계산할 값을 입력하세요 ==> 100

두 번째 계산할 값을 입력하세요 ==> 200

세 번째 계산할 값을 입력하세요 ==> 300

네 번째 계산할 값을 입력하세요 ==> 400

$100 + 200 + 300 + 400 = 1000$

[예제모음 01] 숫자 4개를 더하는 프로그램

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b, c, d;           —— 입력받을 변수 4개를 선언한다.
06     int result;
07
08     printf("첫 번째 계산할 값을 입력하세요 ==> ");
09     scanf("%d", &a);          —— 변수 a에 들어갈 값을 키보드로
                                직접 입력한다.
10     printf("두 번째 계산할 값을 입력하세요 ==> ");
11     scanf("%d", &b);          —— 변수 b에 들어갈 값을 키보드로
                                직접 입력한다.
12     printf("세 번째 계산할 값을 입력하세요 ==> ");
13     scanf("%d", &c);          —— 변수 c에 들어갈 값을 키보드로
                                직접 입력한다.
14     printf("네 번째 계산할 값을 입력하세요 ==> ");
15     scanf("%d", &d);          —— 변수 d에 들어갈 값을 키보드로
                                직접 입력한다.
16
17     result = a + b + c + d;    —— 변수 a, b, c, d의 값을 모두 더해
                                변수 result에 입력한다.
18
19     printf(" %d + %d + %d + %d = %d \n", a, b, c, d, result);
20 }
```

—— 변수 a, b, c, d와 result 값을 모니터에 출력한다.

[예제모음 02] if문을 활용한 계산기

예제 설명 if문으로 덧셈, 뺄셈, 곱셈, 나눗셈 중 하나를 선택하여 계산하는 프로그램이다. 5장에서 배울 if문이 미리 나와서 좀 어렵게 느껴지겠지만 직접 코딩하고 실행해보자.

실행 결과

첫 번째 계산할 값을 입력하세요 ==> 1000

<1>덧셈 <2>뺄셈 <3>곱셈 <4>나눗셈 ==> 2

두 번째 계산할 값을 입력하세요 ==> 122

1000 - 122 = 878

[예제모음 02] if문을 활용한 계산기

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     int result;
07     int k;          ——— 계산 방식을 선택할 변수를 선언한다.
08
09     printf("첫 번째 계산할 값을 입력하세요 ==> ");
10     scanf("%d", &a); ——— 계산할 수를 입력한다.
11     printf("<1>덧셈 <2>뺄셈 <3>곱셈 <4>나눗셈 ==> ");
12     scanf("%d", &k); ——— 연산자를 선택한다(1: 덧셈, 2: 뺄셈,
13     printf("두 번째 계산할 값을 입력하세요 ==> ");      3: 곱셈, 4: 나눗셈).
14     scanf("%d", &b); ——— 계산할 수를 입력한다.
15
16     if(k == 1) { ——— 입력한 k가 1이면 덧셈을 수행한다.
17         result = a + b;
18         printf(" %d + %d = %d \n", a, b, result);
19     }
```

[예제모음 02] if문을 활용한 계산기

20

```
21  if(k == 2) {  
22      result = a - b;  
23      printf(" %d - %d = %d \n", a, b, result);  
24  }
```

—— 입력한 k가 2이면 뺄셈을 수행한다.

25

```
26  if(k == 3) {  
27      result = a * b;  
28      printf(" %d * %d = %d \n", a, b, result);  
29  }
```

—— 입력한 k가 3이면 곱셈을 수행한다.

30

```
31  if(k == 4) {  
32      result = a / b;  
33      printf(" %d / %d = %d \n", a, b, result);  
34  }
```

—— 입력한 k가 4이면 나눗셈을 수행한다.

```
35 }
```

[예제모음 03] 오류가 없는 계산기

예제 설명 [예제모음 02]는 0으로 나누면 오류가 발생하므로, 오류 없는 계산기 프로그램을 작성해보자. 연산자는 기호(+, -, *, /)를 사용해서 직접 입력하고 나머지값 연산자인 %를 추가한다.

실행 결과

첫 번째 계산할 값 ==> 100

+ - * / ==> /

두 번째 계산할 값 ==> 0

0으로 나누면 안됩니다.

[예제모음 03] 오류가 없는 계산기

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     int result;
07     char k;          —— 연산자를 입력받을 변수를 문자형으로 선언한다.
08
09     printf("첫 번째 계산할 값 ==> ");
10     scanf("%d", &a);
11     printf("+ - * / % ==> ");
12     scanf(" %c", &ch); —— %c의 앞에 공백이 있어야 한다.
13     printf("두 번째 계산할 값 ==> ");
14     scanf("%d", &b);
15
16     if(k == '+') {
17         result = a + b;
18         printf(" %d + %d = %d \n", a, b, result);
19     }
```

[예제모음 03] 오류가 없는 계산기

```
20
21  if(k == '-') {
22      result = a - b;
23      printf(" %d - %d = %d \n", a, b, result);
24  }
25
26  if(k == '*') {
27      result = a * b;
28      printf(" %d * %d = %d \n", a, b, result);
29  }
30
31  if(k == '/') {
32      if(b != 0) {
33          result = a / b;
34          printf(" %d / %d = %d \n", a, b, result);
35      } else
36          printf(" 0으로 나누면 안됩니다. \n");
37  }
```

—— 0으로 나누거나 나머지를
구하면 처리하지 않고
오류 메시지를 보여준다.

[예제모음 03] 오류가 없는 계산기

```
38
39  if(k == '%') {
40      if(b != 0) {
41          result = a % b;
42          printf(" %d %% %d = %d \n", a, b, result);
43      } else
44          printf(" 0으로 나누면 나머지값이 안됩니다. \n");
45  }
46 }
```

—— 0으로 나누거나 나머지값을
구하면 처리하지 않고
오류 메시지를 보여준다.

감사합니다!

