7강

함수

동양미래대학교 강환수 교수

본 강의 사용 및 참조 자료

- Perfect C, 3판, 강환수 외 2인 공저, 인피니티북스, 2021

9장 함수 기초

목차

01

함수 정의와 호출

함수의 이해

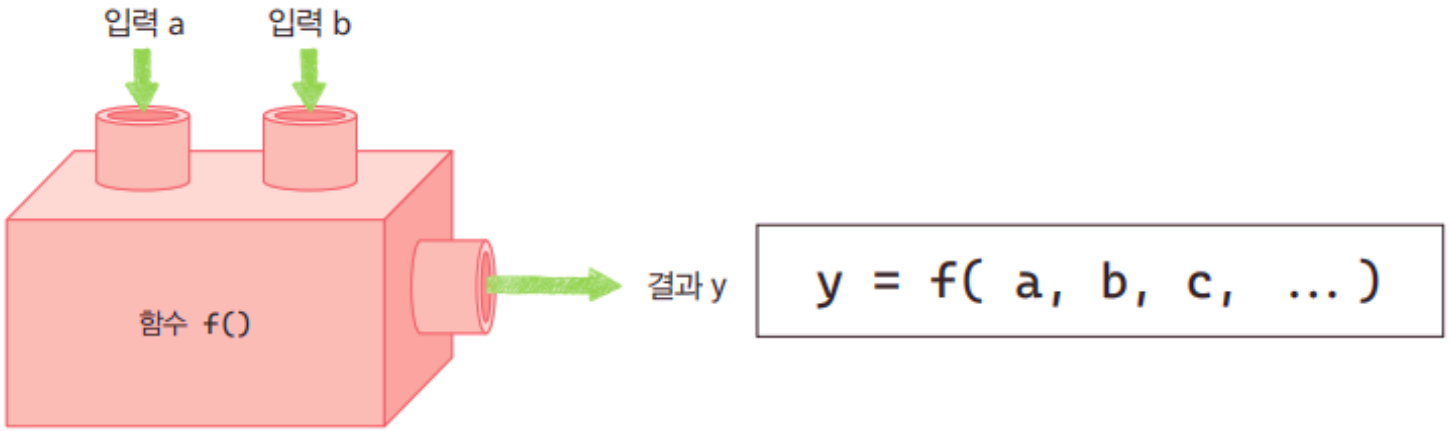
함수 개념

- 함수(function)
  - 프로그램에서 원하는 특정한 작업을 수행하도록 설계된 독립된 프로그램 단위
    - 필요한 입력을 받아 원하는 기능을 수행한 후 결과를 반환(return)
- 라이브러리 함수(library function)와 사용자 정의 함수(user defined function)로 구분
  - 라이브러리 함수
    - 이미 개발환경에 만들어 놓은 라이브러리
  - 사용자 정의 함수
    - 필요에 의해서 개발자가 직접 개발하는 함수

C 프로그램 함수

여러 함수로 구성되는 프로그램

- 최소한 main() 함수와 필요한 다른 함수로 구성되는 프로그램



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 378쪽

C 프로그램 함수

- 함수 main()
  - 이름이 지정된 함수로서
    프로그램의 실행이 시작되는 특수한 함수
  - 첫 줄에서 시작하여 마지막 줄을 마지막으로 실행한 후 종료

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 378쪽

함수 정의와 호출

- 사용자가 직접 개발한 함수를 사용
  - 함수 선언(function declaration)과 함수 호출(function call),
    함수 정의(function definition)가 필요
- 라이브러리 함수
  - 함수 선언(function declaration)과
    함수 정의(function definition)가 이미 구현
  - 함수 호출(function call)로 사용

하나의 프로젝트: 응용프로그램

- 하나의 main() 함수와 여러 개의 다른 함수로 구성
  - 필요에 따라 여러 소스 파일로 나누어 프로그래밍 가능

하나의 응용 프로그램 구성

사용자 정의 함수 호출

```c
#include <stdio.h>

void message(); //함수 원형
void cacao();   //함수 원형

int main(void)
{
   puts("메인 함수 시작입니다.");
   message();    //함수 message 호출
   cacao();      //함수 cacao 호출
   puts("메인 함수 종료입니다.");

   return 0;
}
```

함수 선언

사용자 정의 함수 호출

라이브러리 함수 호출

사용자 정의 함수: 직접 개발한 함수

```c
void message()
{
   puts("\t메시지입니다.");
}
```

함수 정의

사용자 정의 함수: 직접 개발한 함수, 다른 파일도 가능

```c
void cacao()
{
   puts("\t카톡입니다.");
}
```

함수 정의

라이브러리 함수: 개발환경에 이미 등록된 함수

```c
int puts(char *str)
{
   ...
}
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 379쪽

함수 정의 구문 1/3

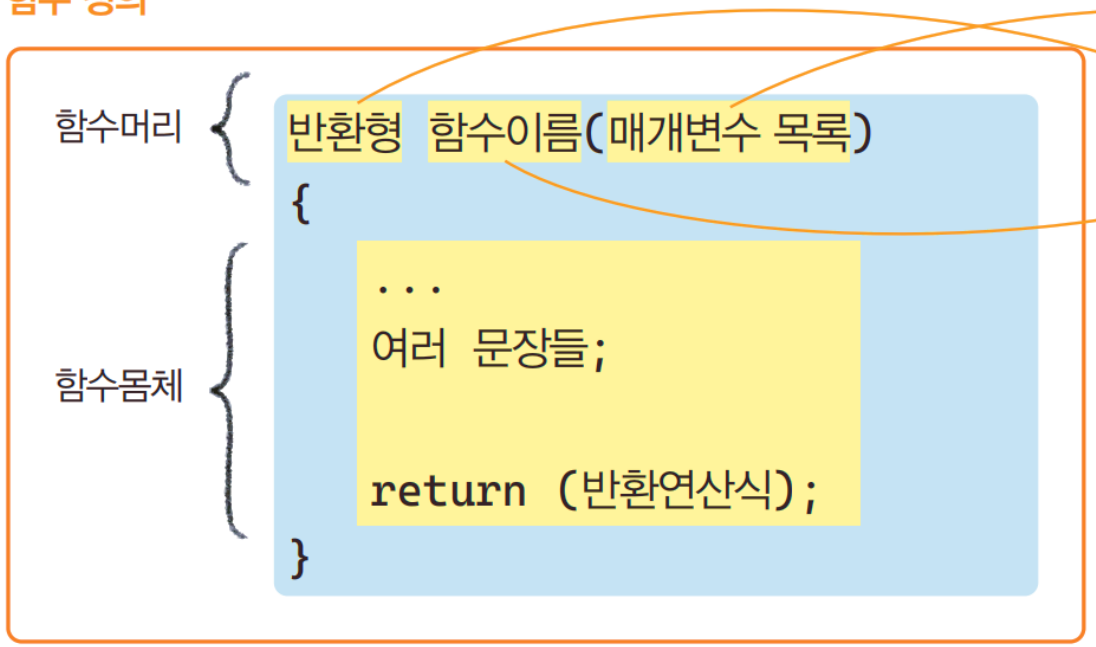
- 함수머리(function header)
  - 반환형과 함수이름, 매개변수 목록으로 구성
  - 반환자료형: 함수 결과값의 자료형
    - 간단히 반환형
  - 함수이름: 식별자의 생성규칙
  - 매개변수 목록: '자료형 변수이름'의 쌍
    - 필요한 수만큼 콤마로 구분하여 기술

함수 정의 구문 2/3

- 함수몸체(function body)
  - {…}와 같이 중괄호로 시작하여 중괄호로 종료
  - 함수가 수행해야 할 문장들로 구성
  - 마지막은 대부분 결과값을 반환하는 return 문장으로 종료
  - 결과값이 없다면 return 생략 가능

함수 정의 구문 3/3

함수 정의




```
int add(int a, int b)
{
    int sum = a + b;

    return (sum);
}
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 382쪽

반환형과 return

- 함수에서 반환값을 전달하는 목적과 함께 함수의 작업 종료를 알리는 문장
  - 함수가 반환값이 없다면 반환형으로 void를 기술

```c
int findMin(int x, int y)
{
    int min = x < y ? x : y;

    return (min);
}
```

```c
void printMin(int a, int b)
{
    int min = a < b ? a : b;
    printf("%n", min);

    return;        //생략 가능
}
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 383쪽

함수원형

- function prototype
  - 함수도 정의된 함수를 호출하기 이전에 필요
  - 함수원형은 함수를 선언하는 문장
- 구문
  - 함수머리에 세미콜론을 넣은 문장
    - int add(int a, int b);
    - int add(int, int);

함수원형 선언 방법

함수원형을 함수 main() 위에 배치

```
#include <stdio.h>

int add(int a, int b);          // 함수 선언
//int add(int, int)도 가능

int main(void)
{
    int a = 3, b = 5;

    int sum = add(a, b);        // 함수 호출
    ...

    return 0;
}

int add(int a, int b)           // 함수 정의
{
    int sum = a + b;
    return (sum);
}
```

함수원형을 함수 main() 내부에 배치

```
#include <stdio.h>

int main(void)
{
    int add(int a, int b);      // 함수 선언
    //int add(int, int)도 가능

    int a = 3, b = 5;

    int sum = add(a, b);        // 함수 호출
    ...

    return 0;
}

int add(int a, int b)           // 함수 정의
{
    int sum = a + b;
    return (sum);
}
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 385쪽

실습예제 1/2

Prj02 02funcadd.c 함수의 정의와 호출 난이도: ★

```c
#include <stdio.h>

//int add(int a, int b);    //이 위치도 가능

int main(void)
{
    int a = 3, b = 5;
    int add(int a, int b);    //int add(int, int)도 가능

    //위 함수원형이 없으면 함수 호출에서 경고 발생
    int sum = add(a, b);
    printf("합: %d\n", sum);

    return 0;
}

```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 386쪽

실습예제 2/2

```
17  //함수 add의 함수 구현 또는 함수 정의 부분
18  int add(int a, int b)
19  {
20      int sum = a + b;
21
22      return (sum); //괄호는 생략 가능
23  }
24
25  //위 main() 함수에서 호출이 없으므로 이 함수 구현은 실행되지 않음
26  int findMin(int x, int y)
27  {
28      int min = x < y ? x : y;
29
30      return (min);
31  }
```

변수 sum의 값을 반환하는 return 문장으로 괄호는 생략 가능하며, 함수 호출한 곳으로 매개변수는 a+b의 결과를 반환

```
합: 8
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 386쪽

02

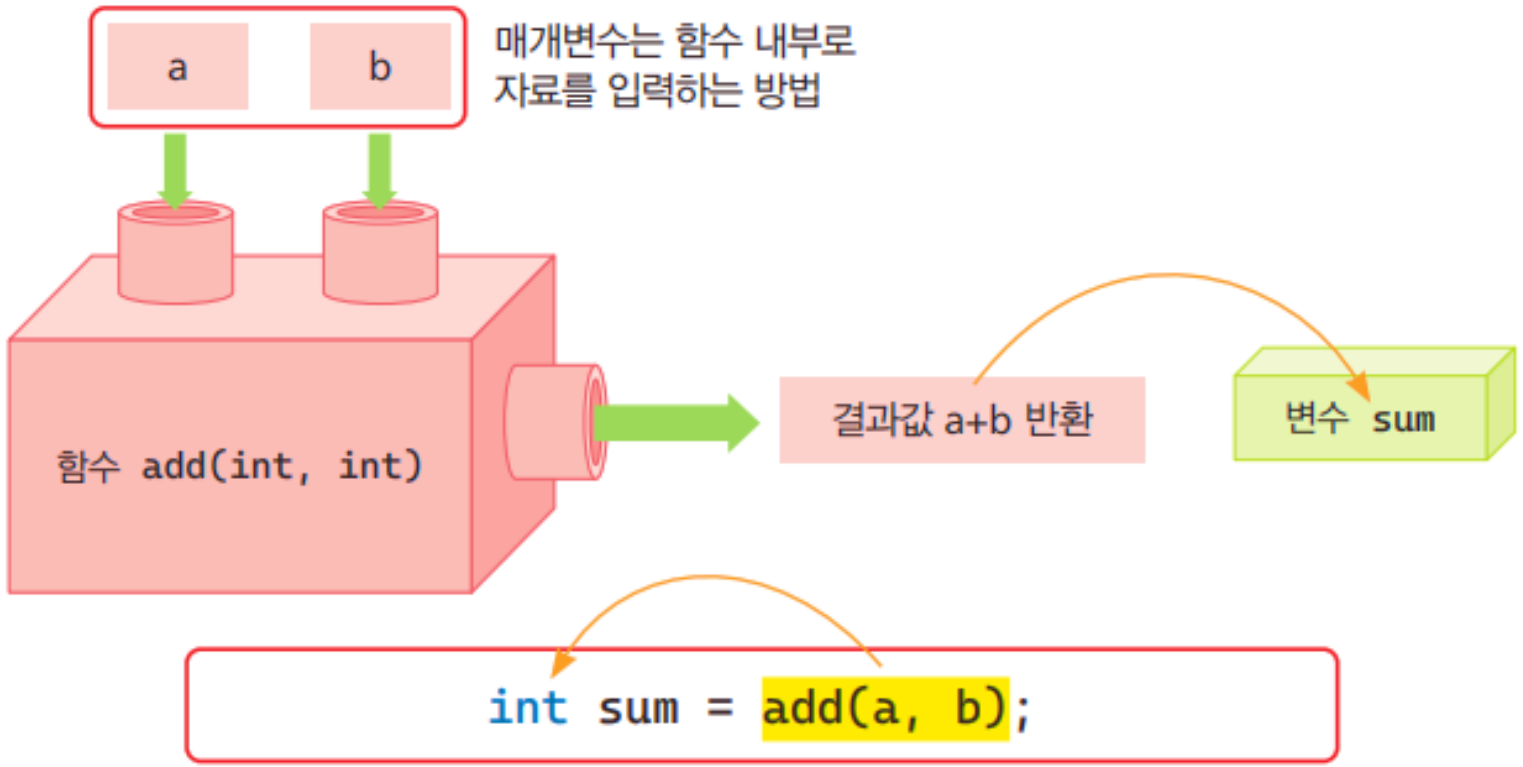
함수 매개변수 활용

매개변수 정의

- 함수 매개변수(parameter)
  - 함수를 호출하는 부분에서 함수몸체로 값을 전달할 목적으로 이용
  - 자료형과 변수명의 목록으로 표시
  - 필요 없으면 키워드 void를 기술
- 반환값
  - 함수를 호출하는 부분에서 함수가 작업을 수행한 후,
    다시 함수를 호출한 영역으로 보내는 결과값

매개변수 목록

여러 개 가능

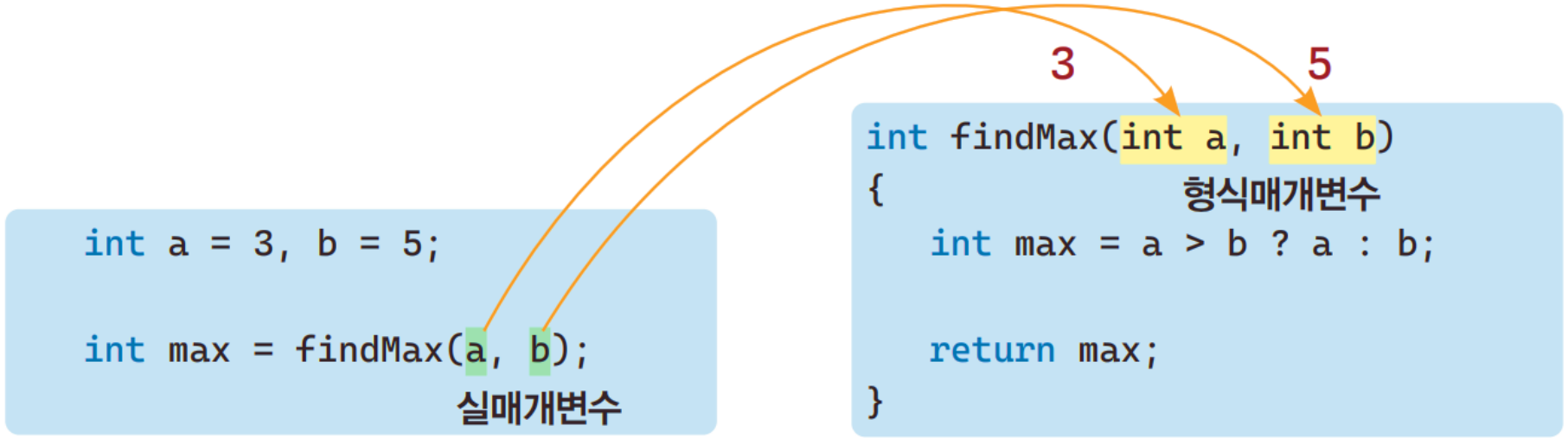


[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 390쪽

형식매개변수와 실매개변수 1/2

- 형식매개변수(formal parameters)
  - 함수 정의에서 기술되는 매개변수 목록의 변수
  - 함수 내부에서만 사용될 수 있는 변수
- 실매개변수(real parameters)와 실인자(real argument)
  - 함수를 호출할 때 기술되는 변수 또는 값
  - 간단히 인자(argument)라고도 부름
- 일반적으로 매개변수, 인자, 인수
  - 구분하지 않고 사용하는 경우도 많음

형식매개변수와 실매개변수 2/2



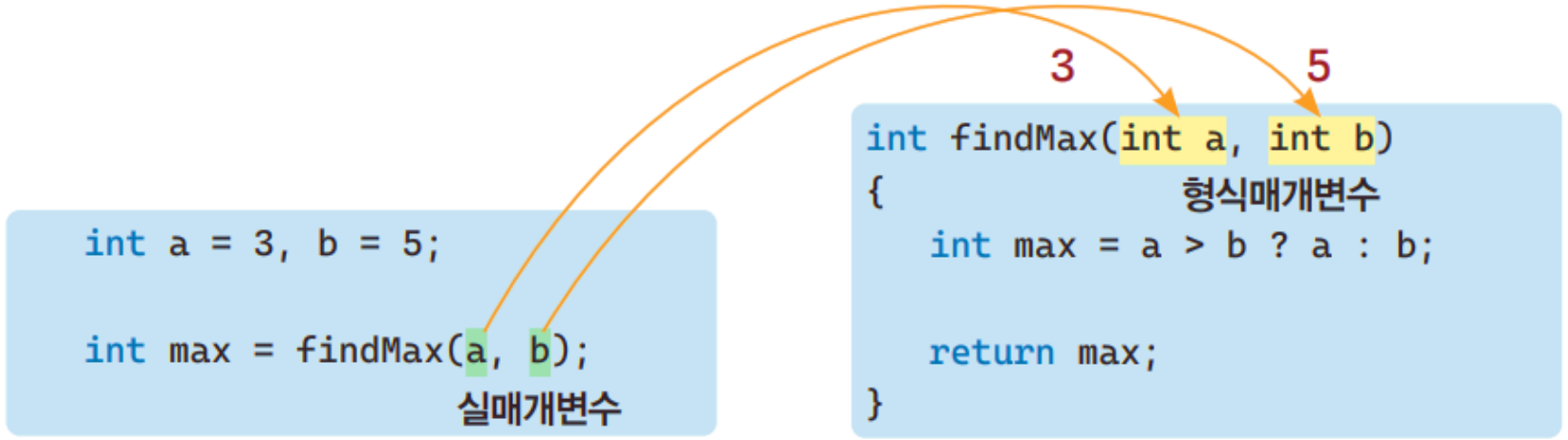
함수 호출 시 매개변수의 수와 자료형이 다르면 문법오류가 발생한다.

```
int max = findMax();          //오류 발생
int max = findMax(2);         //오류 발생
int max = findMax(2.4);       //오류 발생
int max = findMax(2.4, 6.8);  //오류 발생
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 391쪽

값에 의한 호출(call by value)

- 함수가 호출되는 경우 실인자의 값이 형식인자의 변수에 각각 복사된 후 함수가 실행
  - 형식인자 a, b와는 전혀 다른 변수
  - 함수에서 매개변수(일반 변수)의 값을 수정하더라도 함수를 호출한 곳에서의 실제 변수의 값은 변화되지 않는 특징



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 391쪽

실습예제 1/3

하나의 프로젝트가 2개의 소스파일을 소유

Prj03	03functioncall.c 03others.c	함수의 정의와 호출	난이도: ★

03functioncall.c

```
01  #include <stdio.h>
02
03  int add(int a, int b);          //int add(int, int)도 가능
04  int findMax(int, int);          //int findMax(int a, int b)도 가능
05  void printMin(int, int);        //int printMin(int a, int b)도 가능
06
07  int main(void)
08  {
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 391쪽

실습예제 2/3

```
09      int a = 10, b = 15;
10
11      int max = findMax(a, b);
12      printf("최대: %d\n", max);
13      printf("합: %d\n", add(a, b));
14
15      //반환값이 없는 함수 호출은 일반문장처럼 사용
16      printMin(a, b);
17
18      return 0;
19  }
20
21  void printMin(int a, int b)
22  {
23      int min = a < b ? a : b;
24      printf("최소: %d\n", min);
25  }
```

함수 printMin()을 호출하여 a와 b 중에서 작은 값을 함수 정의에서 바로 출력하는데, printMin()은 반환값이 없으므로 대입문의 오른쪽에 r-value로 사용할 수 없음

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 392쪽

실습예제 3/3

03others.c

```c
//함수 add, findMax, findMin, printMin 구현
int add(int a, int b)
{
   int sum = a + b;

   return (sum);
}

int findMax(int a, int b)
{
   int max = a > b ? a : b;

   return max;
}

int findMin(int x, int y)
{
   int min = x < y ? x : y;

   return (min);
}
```

```
최대: 15
합: 25
최소: 10
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 392쪽

03

재귀와 라이브러리 함수

재귀 함수와 재귀적 특성 1/2

- 재귀 함수(recursive function)
  - 함수구현에서 자신의 함수를 호출하는 함수
- 재귀적 특성을 표현하는 알고리즘
  - 재귀 함수를 이용하면 문제를 쉽게 해결

재귀 함수와 재귀적 특성 2/2

- n의 계승(n factorial)을 나타내는 수식
  - n! = 1 * 2 * 3 * ... * (n-2) * (n-1) * n
- n!의 정의에서 재귀적 특성
  - n!을 정의하는 데 (n-1)!을 사용

$$n! \begin{cases} 0! = 1 \\ \\ n! = n * (n-1)! \quad \text{for } (n >= 1) \end{cases}$$

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 401쪽

재귀 함수 구현

```
if  (n <= 1)
      n! = 1
else
      n! = n * (n-1)!
```

```
int factorial(int num)
{
   if (num <= 1)
      return 1;
   else
      return (num * factorial(num - 1));
}
```

$$n! \begin{cases} 0! = 1 \\ n! = n * (n-1)! \quad for\ (n >= 1) \end{cases}$$

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 401쪽

실습예제

Prj06 06factorial.c n!을 구현한 재귀함수 난이도: ★

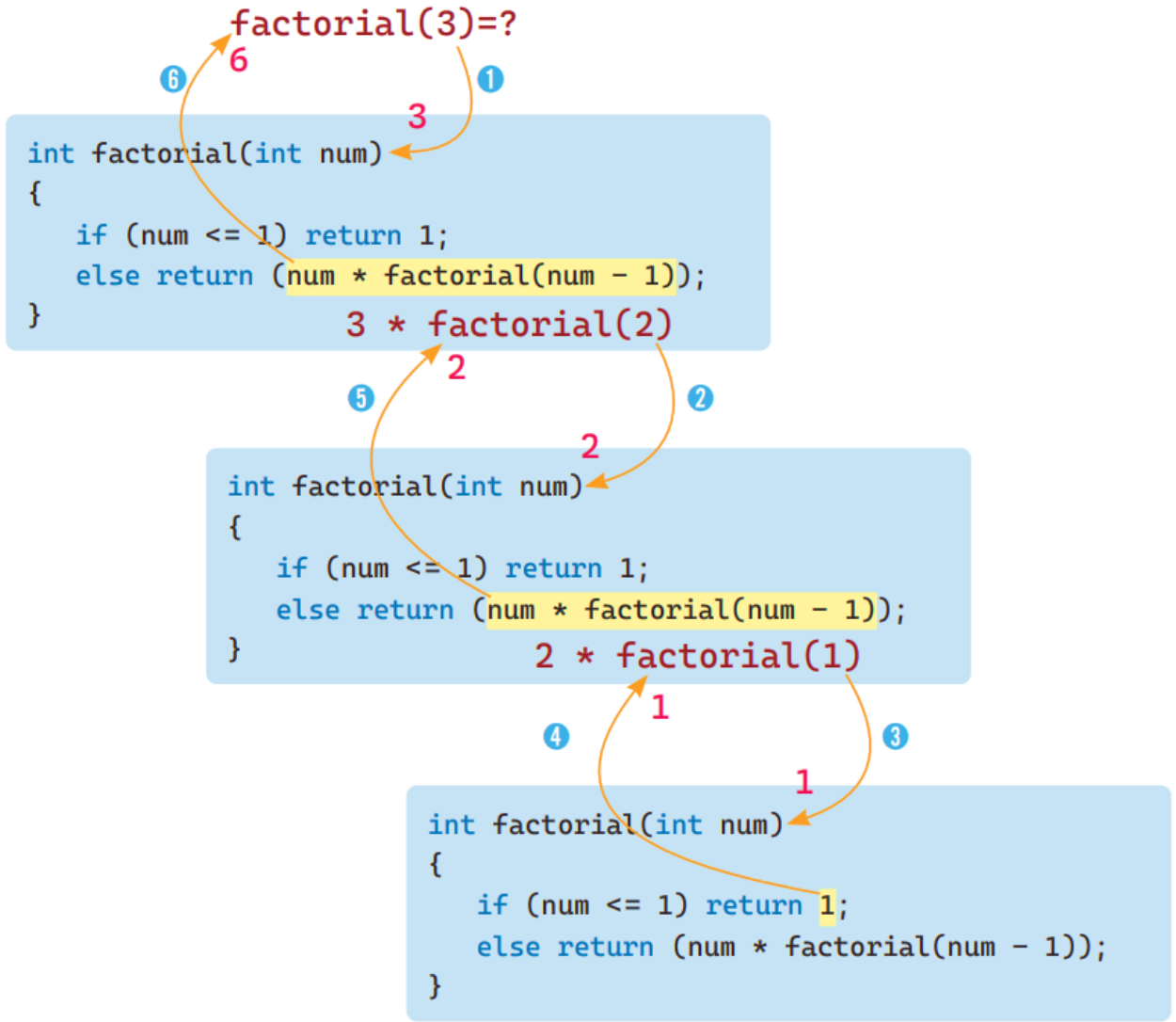
```
01  #include <stdio.h>
02
03  int factorial(int); //함수원형
04
05  int main(void)
06  {
07     for (int i = 1; i <= 10; i++)
08        printf("%2d! = %d\n", i, factorial(i));
09
10     return 0;
11  }
12
13  // n! 구하는 재귀함수
14  int factorial(int number)
15  {
16     if (number <= 1)
17        return 1;
18     else
19        return (number * factorial(number - 1));
20  }
```

조건식을 만족하면 더 이상 자기 자신인 함수 factorial()을 호출하지 않는다.

```
 1! = 1
 2! = 2
 3! = 6
 4! = 24
 5! = 120
 6! = 720
 7! = 5040
 8! = 40320
 9! = 362880
10! = 3628800
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 402쪽

재귀함수 실행



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 403쪽

난수(random number)

- 특정한 나열 순서나 규칙을 가지지 않는 연속적인 임의의 수
  - 임의의 수란 어느 수가 반환될 지 예측 불가
  - 어느 수가 선택될 확률이 모두 동일하다는 의미
- 난수를 발생시키는 장치
  - 주사위나 로또 복권 당첨기가 같은 기구

난수를 생성하는 함수 rand()를 이용

- 로또 프로그램 등 임의의 수가 필요로 하는 다양한 프로그래밍에 활용
  - 함수원형은 헤더파일 stdlib.h(standard library)에 정의
  - 0에서 32767사이의 정수 중에서 임의로 하나의 정수를 반환

```
#include <stdlib.h> //rand() 위한 헤더파일 포함

int main(void)
{

    ...
      printf("%5d ", rand());

}
```

함수 rand()를 사용하려면 헤더 파일 stdlib.h를 삽입해야 한다.

함수 rand()는 0에서 32767까지의 정수 중에서 하나를 반환한다

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 404쪽

실습예제

Prj07 07rand.c 난수를 위한 함수 rand()의 이용 난이도: ★

```
01  #include <stdio.h>
02  #include <stdlib.h> //rand()를 위한 헤더파일 포함
03
04  int main(void)
05  {
06      printf("0 ~ %5d 사이의 난수 8개: rand()\n", RAND_MAX);
07      for (int i = 0; i < 8; i++)
08          printf("%5d ", rand());
09      puts("");
10
11      return 0;
```

기호 상수 RAND_MAX를 16진수 0x7fff로 정의

```
0 ~ 32767 사이의 난수 8개: rand()
   41 18467  6334 26500 19169 15724 11478 29358
```

여러 번 실행에도 항상 같은 수가 출력

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 404쪽

실습예제 1/2

- 수학 관련 라이브러리 함수를 사용하려면 헤더파일 math.h을 삽입

```
  i    i제곱    i세제곱   제곱근(sqrt)
-------------------------------------
  3     9.0      27.0        1.7
  4    16.0      64.0        2.0
  5    25.0     125.0        2.2
  6    36.0     216.0        2.4

2.72,  3.14,  9.00
4.00,  5.00, 10.20
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 407쪽

실습예제 2/2

Prj09 09math.c 다양한 수학 관련 함수 난이도: ★

```c
#include <stdio.h>
#include <math.h> //수학 관련 다양한 함수머리 포함 헤더파일

int main(void)
{
   printf("  i    i제곱    i세제곱   제곱근(sqrt)\n");
   printf("-----------------------------------\n");
   for (int i = 3; i < 7; i++)
      printf("%3d %7.1f %9.1f %9.1f\n", i, pow(i, 2), pow(i, 3), sqrt(i));
   printf("\n");

   printf("%5.2f, ", exp(1.0));
   printf("%5.2f, ", pow(3.14, 1.0));
   printf("%5.2f\n", sqrt(81));
   printf("%5.2f, ", ceil(3.6));
   printf("%5.2f, ", floor(5.8));
   printf("%5.2f\n, ", fabs(-10.2));

   return 0;
}
```

함수 sqrt(81)는 $\sqrt{81}$ 반환

함수 ceil(3.6)은 3.6의 천정 값인 4.0을 반환하는데,
천정 값 ceil(x)이란 x보다 작지 않은 가장 작은 정수를 뜻함

floor(5.8)은 5.8의 바닥 값인 5.0을 반환하는데,
바닥 값 floor(x)이란 x보다 크지 않은 가장 큰 정수를 뜻함

```
  i    i제곱    i세제곱   제곱근(sqrt)
-----------------------------------
  3     9.0      27.0      1.7
  4    16.0      64.0      2.0
  5    25.0     125.0      2.2
  6    36.0     216.0      2.4

2.72,  3.14,  9.00
4.00,  5.00, 10.20
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 407쪽

다양한 라이브러리 함수를 제공

여러 헤더파일이 제공

- 여러 헤더파일에 나뉘어 있음
  - 여러 라이브러리 함수를 위한 함수원형과 상수, 매크로

헤더파일	처리 작업
stdio.h	표준 입출력 작업
math.h	수학 관련 작업
string.h	문자열 작업
time.h	시간 작업
ctype.h	문자 관련 작업
stdlib.h	여러 유틸리티(텍스트를 수로 변환, 난수, 메모리 할당 등) 함수

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 410쪽

정 리 하 기

정리하기

- 특정 작업을 수행하는 프로그래밍 단위인 함수를 이해한다.
- C 프로그램에서 함수를 구현하고 사용한다.
- C 프로그램에서 함수원형, 함수정의, 함수호출을 활용한다.
- 함수에서 함수 내부로 자료를 전달하는 매개변수를 활용할 수 있다.
- 재귀함수를 이해하고
  재귀특성을 이용한 재귀함수를 구현할 수 있다.
- C 언어의 라이브러리 함수를 이해하고
  적절한 헤더파일을 사용해 함수를 호출한다.

다음시간 안내

8강

01~07강 요약