

개정판

쉽게 풀어쓴 C언어 Express

천인국 지음

홈페이지(www.booksr.co.kr) 제공

프로그램 소스, 중간 점검 해답



쉽고 재미있는 프로그래밍의 세계로!

- 다양하고 효과적인 그림으로 내용을 한눈에 이해
- 문답식의 쉬운 설명과 간결하고 적절한 예제로 쉽고 빠른 이해
- 흥미 있는 실습 문제와 단계적인 실습으로 체계적인 심화 학습 가능
- 한눈에 볼 수 있는 프로그램 소스와 소스 설명으로 쉽게 프로그램 이해

머리말

“C언어 익스프레스(개정판)”를 강의 교재로 채택해주셔서 감사드립니다. 본 해답집을 만드는데 나름대로 노력을 기울였으나 제가 가진 지식의 한계로 말미암아 잘못된 부분이 있을 것으로 사료됩니다. 잘못된 부분을 발견하시면 chunik@sch.ac.kr로 연락주시면 더 좋은 책을 만드는데 소중하게 사용하겠습니다. 다시 한 번 감사드립니다.

CHAPTER 1

중간점검문제

p.22

1. 계산기는 수행하는 작업이 고정되어 있으므로 컴퓨터라고 할 수 없다.
2. 이진수로 되어 있는 기계어
3. 고급 프로그래밍 언어를 기계어로 번역한다.

p.25

1. FORTRAN
2. PASCAL
3. C++
4. COBOL
5. JAVA
6. C++, JAVA

p.28

1. 임베디드 시스템(embedded system)은 일종의 컴퓨터 시스템으로 한 가지 기능만 수행되도록 설계되었으며 하드웨어에 내장하여 특수한 기능만을 수행한다. 임베디드 시스템은 오늘날 일상 생활에 쓰이는 많은 장치들을 제어하고 있다.

2. 장점: 간결하고 효율적이며 이식성이 뛰어나다.

단점: 배우기 어렵다. 객체 지향 프로그래밍이 힘들다.

p.32

1. 수화기를 든다. -> 상대방의 전화번호를 누른다. -> 통화를 한다 -> 통화가 끝나면 수화기를 내려 놓는다.
2. 세탁기의 전원을 켜다. -> 세탁기의 뚜껑을 연다. -> 옷과 세제를 넣는다. ->세탁기의 뚜껑을 닫는다. ->실행 버튼을 누른다. -> 완료되었으면 전원을 차단한다.

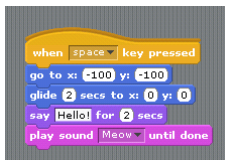
p.39

1. 요구사항분석->알고리즘 작성->소스 작성->컴파일과 링크->프로그램 실행과 디버깅
2. 올바르지 않다. 확장자로 .c를 붙여야 한다.
3. 소스 파일: 프로그래밍 언어로 작성한 원본 파일
오브젝트 파일: 소스 파일을 기계어로 컴파일한 파일
실행파일: 오브젝트 파일들을 모아서 시동 코드와 라이브러리 코드를 붙인 파일
4. 오브젝트 파일: **test.obj** 실행파일: **test.exe**
5. 소스 파일은 반드시 보관하여야 한다. 오브젝트 파일은 삭제하여도 된다.
6. 원하는 실행 결과가 나오지 않는 경우에 “소스 수정->컴파일->링크->실행”의 단계를 다시 거치는 오류 수정 작업
7. 오류가 남아있을 수도 있고 사용자의 요구조건이 변경될 수 있다.

p.43



1.



2.

와 같은 입력하여서 실행하여 본다.

“조건부실행”과 “변수”에서는 변수 생성이 필요하다. 변수는 아직 학습하지 않았지만 값을



저장하고 있는 박스라고 생각할 수 있다. 변수는 클릭한다.

에서 “Make a variable”을

연습문제

1. (1) 이진법
2. (2) 컴파일러
3. (2) 순서도
4. (3)->(2)->(5)->(1)->(6)->(4)

5. (2) 객체 지향 프로그래밍이 가능하다.

6. (1) 알고리즘

7. (1), (2), (3)

8. (2)

9. (1)

10. 이미지의 경우, 각 픽셀의 밝기값이 숫자로 표현된다. 음악의 경우, 음파의 높이가 숫자로 표현된다. 좀 더 자세한 내용을 인터넷에서 조사하여 본다.

11. * 아날로그 방식의 장점: 대용량의 저장이 가능하다.

* 아날로그 방식의 단점: 잡음이나 왜곡에 약하다. 조작이나 변경이 어렵다.

* 디지털 방식의 장점: 잡음에 강하다. 고기능의 추구가 가능하다. 소형화, 저가격화가 가능하다. 멀티미디어가 가능하게 되었다.

* 디지털 방식의 단점: 값을 표현하는 비트의 개수가 적은 경우에, 화질이나 음질이 떨어진다.

12. 0과 9까지를 사용하는 십진법에 비하여 0과 1만을 사용하는 이진법이 표현하기가 쉽기 때문이다. 즉 이진법은 간단한 스위치를 이용하여 0과 1을 표현할 수 있다. 이 스위치들은 전자 회로를 이용하여 간단하게 구현할 수 있다.

13. * MOV(Move): 데이터 이동 (전송)

* XCHG(Exchange Register/memory with Register): 첫 번째 오퍼랜드와 두 번째 오퍼랜드 교환

* IN(Input from AL/AX to Fixed port): 오퍼랜드로 지시된 포트로부터 AX에 데이터 입력

14. 고대의 기계들은 정해진 기능만을 수행하는 반면에, 현대의 컴퓨터는 범용적인 기계로서 프로그램만 변경하면 다양한 기능을 수행할 수 있다.

15. * 컴퓨터 내부 부품 : CPU, RAM, 마더보드, 하드디스크, 그래픽카드, 케이스/파워서플라이, 시디롬, 플로피디스크드라이브, 사운드카드, 네트워크 카드(랜카드), 시디레코더, DVD 드라이브

* 컴퓨터 외부 부품 : 모니터, 키보드, 마우스, 스캐너, 프린터

16. 인간이 문제를 해결하는 방법을 고안하여 한번만 알려주면 컴퓨터는 반복하여 불평없이 작업을 할 수 있기 때문이다.

17. * 기계어 : 특정 컴퓨터의 명령어를 이진수로 표시한 것이며 컴퓨터 하드웨어를 설계할 때 결정된다.

* 어셈블리어 : CPU의 명령어들을 이진수가 아닌 영어의 약자로 표기한 것이다.

* 고급언어 : 특정한 컴퓨터의 구조나 프로세서에 무관하게 독립적으로 프로그램을 작성할 수 있는 언어

18. 객체 지향이란 실제 세계가 구성되어 있는 것과 비슷하게 소프트웨어도 설계하자는 것이다. 우리가 살고 있는 실제 세계는 여러 개의 객체(object)로 이루어져 있다. 객체란 실제 세계 또는 추상적으로 존재하는 사물을 객체(object)라고 한다.

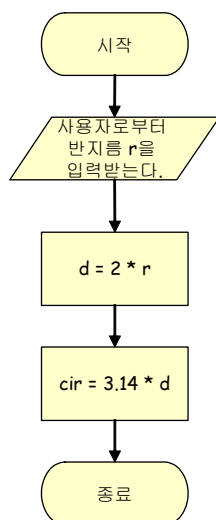
19. 미리 정해진 특정 기능을 수행하기 위해 컴퓨터의 하드웨어와 소프트웨어가 조합된 전자 제어 시스템을 말한다.

20. 아이폰은 Objective-C , 안드로이드는 Java, 윈도우폰은 C# 언어를 사용한다. 3가지 언어 모두 객체 지향 언어이다.

21. (a) 프린터의 전원을 켜다->종이를 넣는다.->인쇄 버튼을 누른다.
(b) 상품을 선택한다->배송지를 입력한다.->결제를 한다

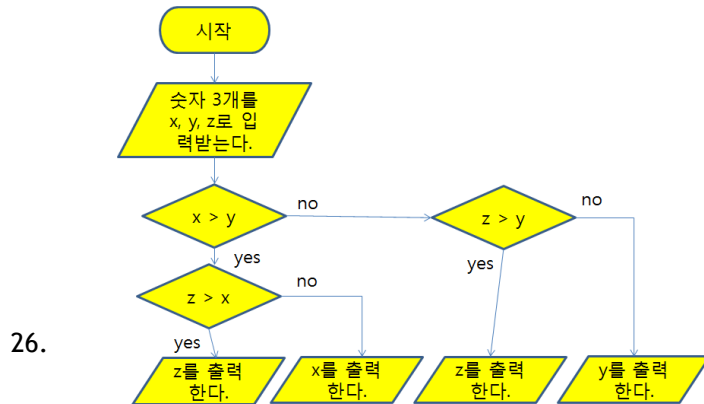
22. 사용자에게 두수를 입력받는다.-> 두수를 더하여 합을 sum에 저장한다.-> sum을 2로 나누어서 화면에 표시한다.

23.

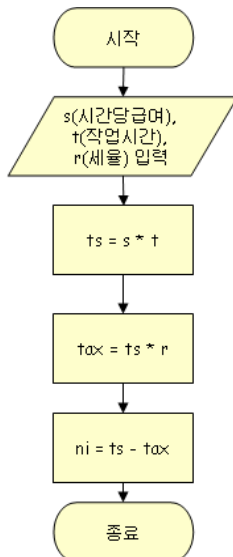


24. 두 사람이 두 장의 카드를 뽑는다. -> 가장 큰 숫자의 카드를 비교한다. -> 더 큰 사람이 이기고 만약에 동일하면 나머지 한 장을 서로 비교한다. -> 더 큰 사람이 이기고 두 번째 카드도 같은 수일 경우 비기게 된다.

25. 10개의 숫자 중 가장 작은 숫자를 제일 첫 번째 있는 숫자와 위치를 바꾼다. -> 첫 번째 숫자를 제외하고 두 번째부터 10번째까지의 숫자 중 가장 작은 숫자를 두 번째 숫자와 위치를 바꾼다. -> 3번째부터 10번째까지 숫자 중 가장 작은 숫자를 3번째 숫자와 위치를 바꾼다. ->...-> 9번째부터 10번째까지 숫자 중 작은 숫자를 9번째 숫자와 위치를 바꾼다.



27.



28. 첫 번째 컵에 우유가 담겨져 있고 두 번째 컵에 주스가 담겨져 있다고 가정한다. -> 첫 번째 컵에 있는 것을 세 번째 컵에 옮긴다. -> 두 번째 컵에 있는 것을 첫 번째 컵에 옮긴다. -> 세 번째 컵에 있는 것을 두 번째 컵에 옮긴다.

29. (a) 숫자들의 목록의 처음에 있는 숫자와 찾고자 하는 숫자를 비교한다. -> 일치하지 않으면 다음 숫자와 비교한다. -> , ..., -> 끝에 있는 숫자와 비교하여 일치하지 않으면 탐색 실패

(b) 숫자들의 목록의 중간에 있는 숫자와 비교한다. -> 찾는 숫자가 중간에 있는 숫자보다 크면 뒤쪽에서 다시 반을 잘라 검색을 하고, 찾는 숫자보다 작으면 앞쪽에서 다시 반을 잘라 검색을 한다.-> 이런식으로 반복하여 숫자를 찾는다.

30. * 문법적인 오류 : The boy, **whom(->who)** was thought to be absent-minded, really had a very active mind.

* 논리적인 오류 : "This statement is false"

31. 차후에 생길지도모르는 버그 수정, 사용자의 추가 요구 사항 충족

32.

- (1) 사용자로부터 정수 x 를 입력받는다.
- (2) 정수 i 에 1을 대입한다.
- (3) 정수 n 에 0을 대입한다.
- (4) x 를 i 로 나눈다. 나누어 떨어지면 약수의 개수 n 을 하나 증가한다.
- (5) i 를 하나 증가한다.
- (6) i 가 x 보다 작거나 같으면 (4)로 돌아간다.
- (7) 약수의 개수 n 이 2이면 소수이다.

CHAPTER 2

중간점검문제

p.43

1. 통합 개발 환경
2. [파일]->[새로만들기]->[프로젝트]
3. [빌드]->[솔루션빌드]
4. 구별한다.
5. sample.obj와 sample.exe
6. 사용하여도 된다. 메모장으로 소스 파일을 생성한 후에 [소스파일] 위에서 마우스 오른쪽 버튼을 눌러서 [추가]->[기존항목]을 선택한다.

p.64

1. 세미콜론(;)
2. 주석(comment)
3. 화면에 주어진 문자열을 출력한다.

p.66

1. \n
- 2.

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

```
int main(void)
```

```
{
```

```
    printf("사과 \n");
```

```
    printf("오렌지 \n");
```

```
    printf("포도 \n");
```

```
    return 0;
```

```
}
```

- 3.

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

```
int main(void)
```

```

{
    printf("3 X 1 = 3 \n");
    printf("3 X 2 = 6 \n");
    printf("3 X 3 = 9 \n");
    printf("3 X 4 = 12 \n");
    printf("3 X 5 = 15 \n");
    printf("3 X 6 = 18 \n");
    printf("3 X 7 = 21 \n");
    printf("3 X 8 = 24 \n");
    printf("3 X 9 = 27 \n");
    return 0;
}

```

p.66

1. 소스 파일을 수정하고 다시 컴파일하여서 실행하는 디버깅(debugging)을 하여야 한다.
2. 경고(warning)

Summary

이름: test -> _test.c -> test.exe

- 각 과정에서 필요한 Visual C++의 메뉴를 빈칸에 적어보자.
메뉴: [파일]>[새로만들기]>[프로젝트]->[파일]>[새로만들기]>[파일]->[빌드]>[솔루션빌드]->[디버그]>[디버깅하지 않고 시작]->[디버그]>[디버깅 시작]
- 다음은 이번 장에서 학습한 소스이다. 소스를 설명하는 설명문을 빈칸에 적어보자.

```

#include <stdio.h>
int main(void)
{
    printf("Hello World! \n");
    return 0;
}

```

// 헤더 파일을 포함시킨다.

main() 함수 선언

화면에 "Hello World!"를 출력한다. 줄을 바꾼다.

함수의 시작과 끝을 표시한다.

연습문제

1. Visual C++로

`printf("C언어를 시작합니다. \n");`

2. (a) `print(Hello World!);` -> `printf("Hello World!");`
(b) `int main(void);` -> `int main(void)`
(c) `int Main(void)` -> `int main(void)`
(d) `print("Hello World!\n");` -> `printf("Hello World!\n");`

3.(a) `/* 홍길동, 20080101, 2008/3/1 */`

(b) `printf("Hi Programmers \n");`

(c) `printf("\n Good Morning");`

(d)

`printf("C 언어 \n");`

`printf("C++ 언어 \n");`

`printf("Java 언어 \n");`

프로그래밍

1.

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

```
int main(void)
```

```
{
```

```
    printf("이름: 홍길동\n나이: 21살\n주소: 서울 200번지\n");
```

```
    return 0;
```

```
}
```

2.

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

```
int main(void)
```

```
{
```

```
    printf("Hello\nC\nProgrammers!\n");
```

```
    return 0;
```

```
}
```

3.

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

```
int main(void)
```

```
{
```

```
    printf("*****\n");
    printf("          리포트          *\n");
    printf("*****\n\n");
    printf("\t-----\n");
    printf("\t|      컴퓨터학과      |\n");
    printf("\t-----\n");
    printf("\t20070001\n");
    printf("\t홍길동\n");
    return 0;
```

```
}
```

4.

(a)

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

```
int main(void)
```

```
{
```

```
    printf("*****\n");
    printf("      *\n");
    printf("      *\n");
    printf("*****\n");
    return 0;
```

```
}
```

(b)

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

```
int main(void)
```

```
{
```

```
    printf("*****\n");
    printf("      *\n");
    printf("      *\n");
    printf("      *\n");
    printf("*****\n");
    return 0;
```

```
}
```

(c)

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

```
int main(void)
{
    printf("  *  \n");
    printf(" * *  \n");
    printf(" *  * \n");
    printf("*    *\n");
    printf(" *  * \n");
    printf(" * *  \n");
    printf("  *  \n");
    return 0;
}
```

CHAPTER 3

중간점검문제

p.84

1. /* /* ... */ */와 같이 중첩할 수 없다.
2. /* ... */ 형태의 주석은 한줄 이상으로 할 수 있다.
3. 코드를 작성한 의도를 적는 것이 좋다.
4. 전혀 영향을 끼치지 않는다.

p.85

1. `stdio.h`
2. 현재의 위치에 헤더파일을 읽어서 포함시키라는 의미이다.

p.88

1. `main()`
2. {와 }이다.
3. ;

p.91

1. `int i;`
2. `double f;`
3. 함수의 첫 부분에서(즉 일반 문장을 작성하기 전에)

p.93

1. `product = a * b;`
2. `quotient = a / b;`

p.96

1. `%f`
2. `printf("k=%d \n", k);`

p.100

```
1. double value;  
scanf("%lf", &value);
```

도전문제

p.103

```
(1) printf("사각형의 넓이: %lf \n사각형의 둘레: %lf\n", area, perimeter);  
(2) double w, h, area, perimeter;  
(3) printf("사각형의 가로: ");  
scanf("%lf", &w);  
printf("사각형의 세로: ");  
scanf("%lf", &h);
```

Summary

- 다음의 프로그램에 설명문을 추가하면서 학습한 내용을 정리하여 보자.

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

헤더파일 포함

```
int main(void)
```

변수 선언

```
{
```

```
int x;
```

```
int y;
```

```
int z;
```

```
printf("정수를 입력하시오: ");
```

변수 x에 저장하기

```
scanf("%d", &x);
```

```
z = x * y;
```

x와 y를 곱하여 z에 대입
하기

```
printf("x * y = %d \n", z);
```

z의 값을 정수형식으로
출력하기

```
return 0;
```

```
}
```

* 주석

- * 들여쓰기
- * 변수
- * 문자열
- * '\n'
- * %d
- * scanf()
- * &
- * 대입

연습문제

1. (3)
2. (1), (2)
3. (3)
4. (2)
5. (3)
- 6.

```
#include <stdio.h>    //헤더파일 포함

int main(void)        //main() 함수 선언_____
{
    int x, y;          //변수 선언_____

    scanf("%d %d", &x, &y);    //사용자로부터 값을 받아서 x와 y에 저장
    printf("덧셈: %d\n", x + y);    //x+y의 값을 정수형식으로 출력_____
    printf("뺄셈: %d\n", x - y);    //x-y의 값을 정수형식으로 출력
    printf("곱셈: %d\n", x * y);    //x*y의 값을 정수형식으로 출력
    printf("나눗셈: %d\n", x / y);    //x/y의 값을 정수형식으로 출력
}
```

- 7.

```
#include <stdio.h>
int main(void)
{
    float inch, mm;
    printf("인치 단위로 입력:");
    scanf("%f", &inch);
    mm = inch * 25.4;
    printf("%f 인치= %f mm", inch, mm);
    return 0;
}
```



```
}
```

8.

```
/* 첫번째 프로그램 */  
#include <stdio.h>  
  
int main(void)  
{  
    int x;  
    int y;  
    int prod;  
    scanf("%d", &x);  
    scanf("%d", &y);  
    prod = x*y;  
    printf("곱셈의 결과= %f, prod );  
  
    return 0;  
}
```

```
/* 첫번째 프로그램 */  
#include <stdio.h>  
  
int main(void)  
{  
    int x;  
    int y;  
    int prod;  
    scanf("%d", &x);  
    scanf("%d", &y);  
    prod = x*y;  
    printf("곱셈의 결과= %f", prod );  
}
```

9.

(a)

It`s never too late.

It never rains but it pours

(b)

Love is blind.

(c)

It`s never too late./n It never rains but it pours

(d)

10

(e)

30

(f)

10 + 20 = 30

(g)

10 * 20 = 200

(h)

*

**

프로그래밍 문제

1.

```
#include<stdio.h>
int main(void)
{
    double i, j, k, sum, ave;

    printf("실수를 입력하시오: ");
    scanf("%lf",&i);
    printf("실수를 입력하시오: ");
    scanf("%lf",&j);
    printf("실수를 입력하시오: ");
    scanf("%lf",&k);

    sum = i+j+k;
    ave = sum/3.0;

    printf("합은 %f이고 평균은 %f입니다.\n", sum, ave);
    return 0;
}
```

```
실수를 입력하시오: 2
실수를 입력하시오: 3
실수를 입력하시오: 4
합은 9.000000이고 평균은 3.000000입니다.
계속하려면 아무 키나 누르십시오 . . .
```

2.

```
#include<stdio.h>
int main(void)
{
    double mile, km;

    printf("마일을 입력하시오: ");
    scanf("%lf",&mile);
    km = mile*1609.0;
```

```

    printf("%f마일은 %f미터입니다. \n", mile, km);
    return 0;
}

```

```

마일을 입력하시오: 2
2.000000마일은 3218.000000미터입니다.
계속하려면 아무 키나 누르십시오 . . .

```

3.

```

#include<stdio.h>
int main(void)
{
    double base, height, area;

    printf("삼각형의 밑변: ");
    scanf("%lf",&base);
    printf("삼각형의 높이: ");
    scanf("%lf",&height);
    area = 0.5*base*height;

    printf("삼각형의 넓이: %f \n", area);
    return 0;
}

```

```

삼각형의 밑변: 6
삼각형의 높이: 3
삼각형의 넓이: 9.000000
계속하려면 아무 키나 누르십시오 . . .

```

4.

```

#include<stdio.h>
int main(void)
{
    float f, c;

    printf("화씨값을 입력하시오: ");
    scanf("%f", &f);
    c = 5.0*(f-32.0)/9.0;
    printf("섭씨값은 %f도입니다. \n", c);
    return 0;
}

```

화씨값을 입력하시오: 32
섭씨값은 0.000000도입니다.
계속하려면 아무 키나 누르십시오 . . .

5.

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

```
int main(void)
```

```
{
```

```
    double num, ans;
```

```
    printf("값을 입력하세요 : ");
```

```
    scanf("%lf" , &num);
```

```
    ans = 3.0*num*num+7.0*num+11;
```

```
    printf("다항식의 값은 %f\n", ans);
```

```
    return 0;
```

```
}
```

값을 입력하세요 : 2.0
다항식의 값은 37.000000
계속하려면 아무 키나 누르십시오 . . .

CHAPTER 4

중간점검문제

p.112

1. 변수는 실행 도중에 값을 변경할 수 있는 공간이다. 상수는 실행 도중에 값을 변경할 수 없는 공간이다.
2. 메모리(기억장치)

p.116

1. 알파벳, 숫자, _으로 구성된다. 중간에 공백이 들어갈 수 없다. 대문자와 소문자는 구별된다. 키워드는 식별자로 사용할 수 없다.
2. 알파벳, _
3. 키워드(예약어)

p.120

1. 변수는 반드시 함수의 처음에서 선언되어야 한다.
2. `int x=1, y=0;`

p.128

1. short, int, long
2. 이해하기 쉽고 값의 변경이 쉬어진다.
3. 메모리를 절약하기 위하여
4. 음수를 양수로 해석한다.
5. -32768이 된다.

p.132

1. 근본적인 이유는 뉘셈 연산을 덧셈으로 하려는 것이다. 즉 $3-4 = 3+(-4)$ 와 같이 하려는 것이다. 덧셈 회로만 있으면 덧셈과 뉘셈을 할 수 있으므로 CPU 안의 회로가 간단해진다.
2. 10111101

p.139

빈칸을 채우면서 정리하여 봅시다.

1. float형은 4바이트이고 double형은 8바이트이다.
2. 1.0e25
3. 가수와 지수를 저장하는 공간이 제한되어 있기 때문이다.

p.146

1. 문자도 정수를 이용해서 나타낸다.
2. char
3. printf("\a");

도전문제

p.146

```
#include <stdio.h>
int main(void)
{
    double light_speed = 300000;    // 빛의 속도를 저장하는 변수(300000km/sec)
    double distance = 149600000;    // 태양과 지구 사이 거리를 저장하는 변수
    // 149600000km로 초기화한다.

    double time;                    // 시간을 나타내는 변수
    int itime;                       // 시간을 나타내는 변수

    time = distance / light_speed;  // 거리를 빛의 속도로 나눈다.
    itime = time;

    printf("빛의 속도는 %fkm/s \n", light_speed);
    printf("태양과 지구와의 거리 %fkm \n", distance);
    printf("도달 시간은 %d분 %d초\n", itime/60, itime%60);    // 시간을 출력한다.

    return 0;
}
```

빛의 속도는 300000.000000km/s
태양과 지구와의 거리 149600000.000000km
도달 시간은 8분 18초
계속하려면 아무 키나 누르십시오 . . .

Summary

- 다음의 프로그램에 설명문을 추가하면서 학습한 내용을 정리하여 보자.

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

int main(void)
{
    int number=100L;
    double radius=1.0e-10;
    char ch='z';
    ...
}
```

기호 상수 정의

number 변수 선언하고
long형의 상수 100으로
초기화

radius 변수를 선언하고
실수 상수 1.0e-10로 초
기화

char형 변수 ch를 선언
하고 'z'로 초기화

- * 변수, 상수
- * 알파벳, 숫자, _
- * 키워드
- * short, int, long
- * float, double, long double
- * 오버플로우
- * 언더플로우
- * 0xa
- * 012
- * 2의 보수
- * 이해하기 쉽고 값의 변경이 쉽다.
- * 아스키 코드
- * 이스케이프 시퀀스

시퀀스	의미
\t	탭문자
\a	경고문자
\n	줄바꿈문자
\\	\
\"	"
\'	'

연습 문제

1. (3)

2. char, unsigned char, short, int, long, unsigned, float, double
3. (3)
4. (2), (5)
5. (3)
6. (3),(4)
7. (4)
8. (a) int (b) double (c) float (d) int (e) char
9. (a) 잘못없음
(b) #는 사용할 수 없는 기호
(c) 숫자로 시작할 수 없음
(d) %를 사용할 수 없음
10. (a) 0.3141592e1
(b) 0.716532e3
11. (1), (5)
12. (2), (6)
13. (a) %f 또는 %lf (b) %d (c) %c (d) %f
14. 1.234는 double 형의 상수이기 때문에 float 형의 변수에 대입하면 컴파일 경고가 발생한다.
15.

```
#include <stdio.h>
int main(void)
{
    int x, y = 0;           // int x=0, y=0;
    char grade = "A";      // char grade = 'A';
    double rate = e10;     // double rate = 1.0e10;
    short double profit = 75.0; // double profit = 75.0;
    int salary = 2,000,000; // int salary=2000000;
}
```

프로그래밍 문제

1.

```
#include <stdio.h>
int main(void)
{
    double data;
```



```
printf("실수를 입력하시오);  
scanf("%lf", &data);  
  
printf("실수형식으로는 %f입니다\n", data);  
printf("지수형식으로는 %e입니다\n", data);  
  
return 0;  
}
```

2.

```
#include <stdio.h>  
int main(void)  
{  
    int data;  
  
    printf("16진수 정수를입력하시오);  
    scanf("%x", &data);  
  
    printf("8진수로는%#o입니다\n", data);  
    printf("10진수로는%d입니다\n", data);  
    printf("16진수로는%#x입니다\n", data);  
  
    return 0;  
}
```

3.

```
#include <stdio.h>  
int main(void)  
{  
    int x, y;  
    int tmp;  
  
    x = 10;  
    y = 20;  
  
    printf("x=%d y=%d\n", x, y);  
  
    tmp = x;  
    x = y;  
    y = tmp;
```

```

printf("x=%d y=%d\n", x, y);

return 0;
}

```

4.

```

#include<stdio.h>
int main(void)
{
    double width, length, height;    // 문제에 실수로 입력하라고 되어있음
    double volume;

    printf("상자의 가로 세로 높이를 한번에 입력: ");
    scanf("%lf %lf %lf", &width, &length, &height);
    volume = length*width*height;
    printf("상자의 부피는 %f입니다, volume);
    return 0;
}

```

5.

```

#include<stdio.h>
int main(void)
{
    const double SQMETER_PER_PYEONG=3.3058;
    double pyeong, m;

    printf("평을 입력하세요);
    scanf("%lf",&pyeong);
    m = pyeong*SQMETER_PER_PYEONG;
    printf("%lf평방미터입니다,m);

    return 0;
}

```

6.

```

#include<stdio.h>
int main(void)
{
    double total;
    total = (3.32e-3*9.76e-8)/(9.12e6+1.87e9);
}

```

```

    printf("%lf\n", total);
    return 0;
}

```

7.

```

#include<stdio.h>
int main(void)
{
    double kenergy, mass, speed;

    printf("질량: ");
    scanf("%lf", &mass);
    printf("속도: ");
    scanf("%lf", &speed);

    kenergy = (1.0/2.0)* mass*speed*speed;
    printf("운동에너지: %lf\n", kenergy);
    return 0;
}

```

```

질량: 100
속도: 200
운동에너지: 2000000.000000
계속하려면 아무 키나 누르십시오 . . .

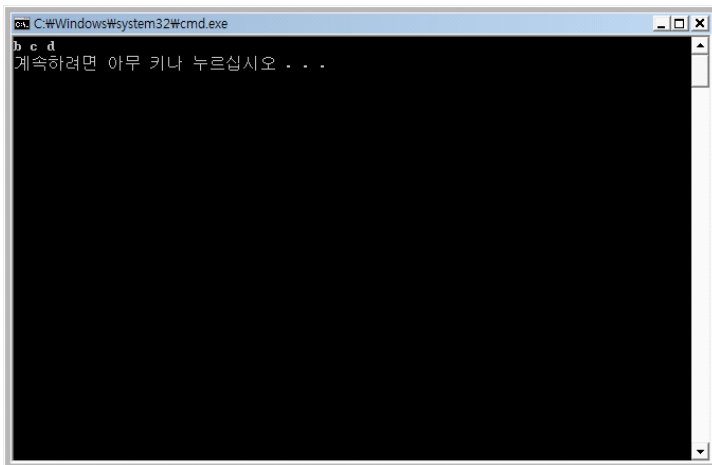
```

8.

```

#include <stdio.h>
int main(void)
{
    printf("%c %c %c\n", 'a'+1, 'a'+2, 'a'+3);
    return 0;
}

```



* 아스키 코드는 알파벳에 연속적인 값을 부여한다.

9.

```
#include <stdio.h>
int main(void)
{
    printf("\a");
    printf("화재가 발생하였습니다.\n");
    return 0;
}
```

10.

```
#include <stdio.h>
int main(void)
{
    printf("\ASCII code", 'A', 'B', 'C\n");
    printf("\t \a \n\n");
    return 0;
}
```

CHAPTER 5

중간점검문제

p.156

1. 수식은 연산자와 피연산자의 조합이다.
2. 상수도 하나의 수식이다.
3. 10과 20은 피연산자이고 +는 연산자이다.
4. 피연산자의 개수

p.162

1. $x = x + 1;$
2. $x++$ 은 x 의 값을 먼저 사용하고 나중에 값을 증가한다. $++x$ 은 증가를 먼저하고 증가된 값을 수식에 사용한다.
3. 13

p.165

1. 변수, 즉 값을 저장할 수 있는 공간이다.
2. 수학에서는 같다는 의미이지만 프로그래밍에서는 저장하라는 의미가 된다.
3. $x = x * y$ 를 줄여서 쓴 것이다.
4. 10을 6으로 나누어서 얻은 나머지 4가 수식의 값이 된다.
5. 10과 6이 모두 정수이므로 결과값은 1이 된다.
6. $x+=y$ 는 $x=x+y$ 와 동일하다. $x+=y$ 는 단순히 $x = +y$ 와 같다. 즉 $x = y$ 와 같다.

p.170

1. 내림 변환은 낮은 등급으로 변환되는 것이고 올림 변환은 높은 등급으로 변환되는 것이다.
2. `(double)x`
3. 정수가 부동소수점수로 형변환되어서 계산이 이루어진다.

p.173

1. 참(1)과 거짓(0)

2. 1+5가 되어서 6이 된다.

p.178

1. (age >= 25 && salary >= 3500)
2. 0이 아니면 참으로 취급한다. 따라서 상수 10은 참이다.
3. !3의 값은 0이 된다.
4. (persons >= 3) && (++count <= 10)
위의 식에서 persons가 3보다 작으면 ++count는 실행되지 않는다.

p.180

1. (x%2==0) ? printf("짝수\n") : printf("홀수\n");

p.186

1. <<
2. ~
3. 원
4. 오른쪽

p.193

1. ,(콤마 연산자)
2. &&
3. 단항연산자
4. 산술연산자

Summary

Summary

빈칸을 채우면서 정리하여 봅시다.

```
#include <stdio.h>
int main(void)
{
    int x=1, y=2, z=3;
    double f=0.0;

    printf("%d\n", 3/5);
    printf("%d\n", 3%5);
    y *= x;
    x++;
    printf("%f\n", 2.0/4);

    x = 12.3;
    f = (double ) x;

    printf("%d\n", 10 > 3);
    printf("%d\n", (10>3) && (1>10));
    printf("%d\n", (10>3) || (1>10));

    x = y = z = 10;

    return 0;
}
```

연습문제

1. (2)
2. (3)
3. (1)
4. (2)
5. (a) 1.5 (b) 1.0 (c) 1.5 (d) 1.5
6. (3)
7. -2, 3, 1, 1
8. 2, 3, 3
9. (a) 0 (b) 0xffff (c) 0xffff (d) x의 모든 비트가 반전된다.
10. (1)
11. 80
12. 1
13.
 - (a) (years >= 3) && (age >= 40) && (families >= 3)
 - (b) (age >=6) && ((height >= 150) || ((height <= 150) && (gaudian == 1)))
 - (c) (gpa >=3.0) && ((toefl >=300) || (toeic>=700))

프로그래밍 문제

```
1.
#include <stdio.h>
int main(void)
{
    int x, y;

    printf("2개의 정수를 입력하시오:");
    scanf("%d %d", &x, &y);

    printf("몫:%d 나머지: %d \n", x/y, x%y);
```



```
    return 0;
}
```

2개의 정수를 입력하시오:5 2
몫:2 나머지: 1

2.

```
#include <stdio.h>
int main(void)
{
    double x, y;
    printf("실수를 입력하시오:");
    scanf("%lf%lf", &x, &y);
    printf("%f %f %f %f \n", x+y, x-y, x*y, x/y);
}
```

3.

```
#include <stdio.h>
int main(void)
{
    int x, y, z, max;
    printf("3개의 정수를 입력하시오:");
    scanf("%d %d %d", &x, &y, &z);
    max = (x > y) ? x : y;
    max = (max > z) ? max : z;
    printf("최대값:%d", max);
}
```

3개의 정수를 입력하시오:2 3 4
최대값:4

4.

```
#include <stdio.h>
int main(void)
{
    int cm;
    int feet;
    double inch;
    const double CM_PER_FEET = (12*2.54);

    printf("키를 입력하시오:");
```

```

scanf("%d", &cm);
feet = cm/(int)(CM_PER_FEET);
inch = (cm -feet*CM_PER_FEET)/2.54;
printf("%d는 %d피트 %f인치입니다.\n", cm, feet, inch);

return 0;
}

```

5.

```

#include <stdio.h>
int main(void)
{
    int value;

    printf("정수를 입력하시오:");
    scanf("%d", &value);
    printf("십의 자리:%d\n", (value/10)%10);
    printf("일의 자리:%d\n", (value/1)%10);
    return 0;
}

```

6.

```

#include <stdio.h>
int main(void)
{
    int value;

    printf("정수를 입력하시오:");
    scanf("%d", &value);
    value = -value;
    value += 0x1;
    printf("2의 보수: %d \n", value);
    return 0;
}

```

7.

```

#include <stdio.h>
int main(void)
{
    int value, count;

    printf("정수를 입력하시오:");

```

```

scanf("%d", &value);
printf("2를 곱하고 싶은 횟수:");
scanf("%d", &count);
printf("%d<<%d의 값: %d \n", value, count, value<<count);
return 0;
}

```

8.

```

#include <stdio.h>
int main(void)
{
    double r, volume, area;
    printf("구의 반지름을 입력하시오: ");
    scanf("%lf", &r);

    area = 4.0*3.141592*r*r;
    volume = (4.0/3.0)*3.141592*r*r*r;
    printf("표면적은 %f입니다.\n", area);
    printf("체적은 %f입니다.\n", volume);
    return 0;
}

```

9.

```

#include <stdio.h>
int main(void)
{
    double stick_height, shadow_length, dist, pyramid_height;
    printf("지팡이의 높이를 입력하시오: ");
    scanf("%lf", &stick_height);
    printf("지팡이 그림자의 길이를 입력하시오: ");
    scanf("%lf", &shadow_length);
    printf("피라미드까지의 거리를 입력하시오: ");
    scanf("%lf", &dist);

    pyramid_height = stick_height*dist/shadow_length;
    printf("피라미드의 높이는 %f입니다.\n", pyramid_height);
    return 0;
}

```

10.

```

#include <stdio.h>

```

```

int main(void)
{
    int x, y;
    printf("x 좌표를 입력하시오: ");
    scanf("%d", &x);
    printf("y 좌표를 입력하시오: ");
    scanf("%d", &y);

    (x>0)?((y>0)?printf("1사분면\n"):printf("4사분면\n")):((y>0)?printf("2사분면\n"):printf("3사분면\n"));

    return 0;
}

```

11.

```

#include <stdio.h>
int main(void)
{
    double dist, degree, circ, radius;

    printf("거리를 입력하시오:");
    scanf("%d", &dist);
    printf("각도를 입력하시오:");
    scanf("%d", &degree);

    circ = (360.0*900.0)/7.2;
    radius = circ/(2.0*3.14);
    printf("지구의 반지름은 %f:", radius);

    return 0;
}

```

```

거리를 입력하시오:900
각도를 입력하시오:7.2
지구의 반지름은 7165.605096

```

12.

```

#include <stdio.h>

int main(void)
{
    unsigned int value;

```

```
char c1, c2, c3, c4;
```

```
printf("첫번째 문자를 입력하시오: ");
```

```
scanf(" %c", &c1);
```

```
printf("두번째 문자를 입력하시오: ");
```

```
scanf(" %c", &c2);
```

```
printf("세번째 문자를 입력하시오: ");
```

```
scanf(" %c", &c3);
```

```
printf("네번째 문자를 입력하시오: ");
```

```
scanf(" %c", &c4);
```

```
value = (c4<<24) | (c3<<16) | (c2<<8) | c1;
```

```
printf("결과값: %x", value);
```

```
}
```

CHAPTER 6

중간 점검 문제

p.208

1. 복합문(블록)
2. 기본적으로 참은 1, 거짓은 0로 표시된다.
3. 관계 수식이나 논리 수식
4. if문 다음에 있는 문장이 실행된다.

```
if (...)  
{  
    ...  
}
```

여기 있는 문장이 실행된다.

5. 문장들을 중괄호로 묶어서 복합문으로 만든다.

p.214

```
1.  
if( n >= 100 )  
    printf("large\n");  
else  
    printf("small\n");
```

p.221

1.
n의 값이 -1인 경우-> C
n의 값이 0인 경우-> A
n의 값이 5인 경우-> B

2.
if(n < 100)
 printf("small\n");

```

else if( n < 200 )
    printf("medium\n");
else
    printf("large\n");

```

p.228

1. 다음 case절의 문장들을 연속하여서 실행한다.

2.

fruit의 값이 1인 경우-> 사과

fruit의 값이 2인 경우-> 배 바나나

fruit의 값이 5인 경우-> 과일

Summary

* 조건문, 반복문

* if(조건식)

문장1; //조건식이 참일 때 실행되는 문장

else

문장2; //조건식이 거짓일 때 실행되는 문장

* 복합문(블록)

switch(조건식) {

case c1: //조건식이 c1과 같으면 실행된다.

문장 1;

break; //switch 외부로 빠져나간다.

case c2: //조건식이 c2과 같으면 실행된다.

문장 2;

break; //switch 외부로 빠져나간다.

...

default : //조건식이 어떤 case절하고도 일치하지 않으면 실행된다.

문장 3;

break;

...

}

연습문제

1. (1)

2. (1)

3. (2), (3)

4.

(a)

```
if( (speed >= 60) && (speed <= 100) )  
    printf("정상 속도 \n");
```

(b)

```
if( x > y ) {  
    max = x;  
    min = y;  
}  
else {  
    max = y;  
    min = x;  
}
```

(b)

```
switch(op) {  
case 1:  
    printf("one");  
    break;  
case 2:  
    printf("two");  
    break;  
case 3:  
    printf("three");  
    break;  
}
```

5.

(a)

```
switch(x)  
{  
case -1:  
    num--;
```



```

        break;
case 1:
    num--;
    break;
default:
    num = 0;
    break;
}

```

(b)

```

if( code == 'X' )
    x++;
else if( code == 'Y' )
    y++;
else
    x = y = 0;

```

6.

(a) if(x>10 && x<20)

```

    printf("%d\n", x);

```

b. if(x<10 || x>20)

```

    printf("%d\n", x);

```

7.

(a)

0

1

(b)

c

8.

(a) if(age>18); -> if(age>18)

(b) if(0 <= age <= 18) -> if(age>= 0 && age <=18)

(c) if(x = 0) -> if(x== 0)

(d)

```

if( speed > 150 )
    printf("벌금 9만원\n");

```

```

else (speed > 150)
    printf("벌금 6만원\n");

```

else

```
printf("OK\n");
```

(e)

```
if( score > 90 )
```

```
    printf("장학금\n");
```

```
    printf("우등\n");
```

```
else
```

```
    printf("좀더 노력하세요\n");
```

->

```
if( score > 90 ) {
```

```
    printf("장학금\n");
```

```
    printf("우등\n");
```

```
}
```

```
else
```

```
    printf("좀더 노력하세요\n");
```

(f)

```
if( x > 0 )
```

```
if( y > 0 )
```

```
    printf("x와 y는 모두 양수\n");
```

```
else
```

```
    printf("x가 양수가 아님.\n");
```

->

```
if( x > 0 )
```

```
if( y > 0 )
```

```
    printf("x와 y는 모두 양수\n");
```

```
else
```

```
    printf("x가 양수가 아님.\n");
```

(g)

```
if( age > 0 | age <= 18 )
```

->

```
if( age > 0 || age <= 18 )
```

(h)

```
int tiger=3;
```

```
switch(animal){
```

```
    case tiger:
```

```
    ...
```

```
}
```

->

```
switch(animal){
```

```
    case 3:
```

```
...  
}
```

프로그래밍 문제

1.

```
#include <stdio.h>  
int main(void)  
{  
    char c;  
    printf("문자를 입력하시오:");  
    c = getchar();  
    switch(c){  
        case 'a':  
        case 'i':  
        case 'o':  
        case 'u':  
        case 'e':  
            printf("모음입니다.\n");  
            break  
        default:  
            printf("자음입니다.\n");  
    }  
    return 0;  
}
```

2.

```
#include <stdio.h>  
int main(void)  
{  
    int x;  
    int y;  
  
    printf("정수를 입력하시오:");  
    scanf("%d", &x);  
    printf("정수를 입력하시오:");  
    scanf("%d", &y);  
}
```

```

        if( x%y == 0 )
            printf("약수입니다.\n");
        else
            printf("약수가 아닙니다.\n");
        return 0;
    }

```

3.

```

#include <stdio.h>
int main(void)
{
    int x, y, z, min;

    printf("3개의 정수를 입력하시오:");
    scanf("%d %d %d", &x, &y, &z);

    if( x < y ){
        if( z < x )
            min = z;
        else
            min = x;
    }
    else {
        if( z < y )
            min = z;
        else
            min = y;
    }
    printf("제일 작은 정수는 %d입니다. \n", min);
    return 0;
}

```

4.

```

#include <stdio.h>
int main(void)
{
    int computer=1;
    int user;
    printf("선택하시오(1: 가위 2:바위 3:보)");
    scanf("%d", &user);
    if( user ==1 && computer ==2)

```

```

        printf("컴퓨터가 이겼음\n");
    else if( user ==1 && computer ==3)
        printf("사용자가 이겼음\n");
    else if( user ==2 && computer ==1)
        printf("사용자가 이겼음\n");
    else if( user ==2 && computer ==3)
        printf("컴퓨터가 이겼음\n");
    else if( user ==3 && computer ==1)
        printf("컴퓨터가 이겼음\n");
    else if( user ==3 && computer ==2)
        printf("사용자가 이겼음\n");
    else
        printf("비겼음\n");

    return 0;
}

```

5.

```

#include <stdio.h>
int main(void)
{
    int height, age;

    printf("키를 입력하시오:");
    scanf("%d", &height);
    printf("나이를 입력하시오:");
    scanf("%d", &age);

    if( height >=140 && age >=10)
        printf("타도 좋습니다.\n");
    else
        printf("죄송합니다. \n");
    return 0;
}

```

6.

```

#include <stdio.h>
int main(void)
{
    int month;

    printf("월번호를 입력하시오: ");

```

```

scanf("%d", &month);
switch(month){
    case 1: printf("Jan\n"); break;
    case 2: printf("Feb\n"); break;
    case 3: printf("Mar\n"); break;
    case 4: printf("Apr\n"); break;
    case 5: printf("May\n"); break;
    case 6: printf("Jun\n"); break;
    case 7: printf("Jul\n"); break;
    case 8: printf("Aug\n"); break;
    case 9: printf("Sep\n"); break;
    case 10: printf("Oct\n"); break;
    case 11: printf("Nov\n"); break;
    case 12: printf("Dev\n"); break;
    default: printf("잘못입력하셨습니다\n"); break;
}
return 0;
}

```

7.

```

#include <stdio.h>
int main(void)
{
    double height, weight, std_weight;

    printf("체중과 키를 입력하세요:");
    scanf("%lf %lf", &weight, &height);
    std_weight = (height-100)*0.9;
    if( weight < std_weight )
        printf("저체중입니다.\n");
    else if( weight > std_weight )
        printf("과체중입니다.\n");
    else
        printf("표준체중입니다.\n");
    return 0;
}

```

8.

```

#include <stdio.h>
int main(void)
{

```

```

int age, time, fee;
printf("현재 시간과 나이를 입력하시오: ");
scanf("%d %d", &time, &age);

if( time < 17 ){
    if( age < 3 )
        fee = 0;
    else if( age >= 65 || age <= 12 )
        fee = 25000;
    else
        fee = 34000;
}
else {
    if( age < 3 )
        fee = 0;
    else
        fee = 10000;
}
printf("요금은 %d입니다.\n", fee);
return 0;
}

```

9.

```

#include <stdio.h>
int main(void)
{
    double x, fx;

    printf("x의 값을 입력하시오:");
    scanf("%lf", &x);
    if( x <= 0 )
        fx = x*x*x - 9.0*x + 2.0;
    else
        fx = 7.0*x+2.0;
    printf("f(x)의 값은 %f\n", fx);
    return 0;
}

```

10.

```

#include <stdio.h>
int main(void)
{

```

```

int x, y;
printf("좌표(x y): ");
scanf("%d %d", &x, &y);
if( x > 0 && y > 0 ){
    printf("1사분면\n");
}
if( x < 0 && y > 0 ){
    printf("2사분면\n");
}
if( x < 0 && y < 0 ){
    printf("3사분면\n");
}
if( x > 0 && y < 0 ){
    printf("4사분면\n");
}

return 0;
}

```

11.

```

#include <stdio.h>
int main(void)
{
    char c;
    printf("문자를 입력하시오: ");
    scanf("%c", &c);
    switch(c) {
        case 'C': printf("Circle \n"); break;
        case 'T': printf("Triangle \n"); break;
        case 'R': printf("Rectangle \n"); break;
        default: printf("Unknown \n"); break;
    }

    return 0;
}

```


CHAPTER 7

중간점검문제

p 242

1. 반복하여 처리하는 작업이 많기 때문이다.
2. while 문, for 문

p,255

1. if문은 조건이 만족되면 한번만 실행한다. while문은 조건이 만족되는 동안 반복하여 실행된다.

2.

```
while(1)
```

```
{
```

```
    //...
```

```
}
```

3.

```
10
```

```
7
```

```
4
```

```
1
```

p,260

1.

```
0
```

```
1
```

```
2
```

p,268

1.

2 4 6 8 계속하려면 아무 키나 누르십시오 . . .

2.

Student10
Student8
Student6
Student4
Student2

p.272

1.

0 곱하기 0은 0
0 곱하기 1은 0
0 곱하기 2은 0
1 곱하기 0은 0
1 곱하기 1은 1
1 곱하기 2은 2
2 곱하기 0은 0
2 곱하기 1은 2
2 곱하기 2은 4
계속하려면 아무 키나 누르십시오 . . .

p.276

1. continue
2. break
- 3.

1 2

4.

1 2 4 5 7 8

연습문제

1.

① 조건식의 값이 1일 때만 참으로 간주된다.-> 조건식의 값이 0이 아니면 참으로 간주된다.

③ **do...while** 문에서 조건식의 값이 거짓이면 한 번도 수행되지 않는다.-> 거짓이라고 하더라도 한번은 수행된다.

④ **for** 문에서 초기식, 조건식, 증감식이 전부 비어 있으면 안 된다.->비어 있어도 된다.

2. ① **while(1) { }** ③ **for(; 1 ;) { }** ④ **for(; ;) { }**

3. 한번도 출력되지 않는다.

4. (a)

0
3
6
9

(b)

0
3
6
9

(c)

0
2
4
6
8

(d)

0
3
6
9

(e)

(f)

54321

5.

(a)

```
int i=10;
while(i>=0){
    printf("%d\n", i);
    i--;
}
```

```
(b)
int i;
for(i=10;i>=0;i-=3)
{
    printf("%d\n",i);
}
```

6. (a) 11 (b) 10

7.

(a) i의 값이 변함이 없으므로 무한루프를 수행하게 된다.

```
int i = 0;
while( i < 10 )
{
    printf("i의 값\n", i);
    i++;
}
```

(b) while이 있는 줄의 끝에 있는 세미콜론(;)을 제거하여야 한다.

```
int i = 0;
while( i++ < 10 )
{
    printf("i의 값\n", i);
}
```

(c) 부등호의 방향을 바꾸어야 한다.

```
int i;
for( i = 0; i < 10; i++ )
    printf("i = %d\n", i);
```

(e) 부동 소수점 수는 오차를 가지고 있으므로 정확히 1.0이 되지 않아서 무한반복이 된다. 만약 1.0까지의 값을 보려 했던 소스라면 $x \neq 1.0$ 을 $x < 1.0$ 으로 수정하는 편이 좋다.

```
float x;
for(x = 0.1; x < 1.0; x += 0.1)
    printf("%f\n", i);
```

(e) 변수 i가 초기화되지 않았다.

```
int i;
```

```
for( i=0 ; i < 10; i++)
    printf("i = %d\n", i);
```

8.

(a)

```
int i;
for(i = 0; i < 5; i++)
{
    printf("%d\n", i);
}
```

(b)

```
int i;
for(i = 0; i < 10 ; i++)
{
    if( i != 5 )
        printf("%d\n", i);
}
```

9. 다음의 수학적식을 계산하는 코드를 작성하라.

(a) $\sum_{i=1}^{30} i^2 + 1$

```
int sum=0, i;
for(i = 1; i <= 30; i++)
    sum += i*i+1;
```

(b) $\sum_{i=10}^{30} \sum_{j=0}^5 (i*j)$

```
int sum=0, i, j;
for(i = 10; i <= 30; i++)
    for(j = 0; j <= 5; j++)
        sum += i * j;
```

프로그래밍 문제

1.

(a)

```
#include <stdio.h>
int main(void)
{
    int i, sum;
    i = 1;
    sum = 0;
    while(i<=100)
    {
        if( (i%3)==0 )
            sum += i;
        i++;
    }
    printf("1부터 100 사이의 모든 3의 배수의 합은 %d입니다. \n", sum);
    return 0;
}
```

(b)

```
#include <stdio.h>
int main(void)
{
    int i, sum;
    sum = 0;
    for(i=1;i<=100;i++)
        if( (i%3)==0 )
            sum += i;
    printf("1부터 100 사이의 모든 3의 배수의 합은 %d입니다. \n", sum);
    return 0;
}
```

(참고) 물론 3에서 시작하여서 3씩 더하여도 된다.

(c)

```
#include <stdio.h>
int main(void)
{
    int i, sum;
    i = 1;
    sum = 0;
    do
    {
        if( (i%3)==0 )
```

```

        sum += i;
        i++;
    } while(i<=100) ;
    printf("1부터 100 사이의 모든 3의 배수의 합은 %d입니다. \n", sum);
    return 0;
}

```

2.

```

#include <stdio.h>
int main(void)
{
    int number, sum;
    sum = 0;
    while (scanf("%d", &number) != EOF )
    {
        sum += number;
    }
    printf("정수의 합은 %d입니다. \n", sum);
    return 0;
}

```

3.

```

#include <stdio.h>
int main(void)
{
    int y, x;

    for(y=0;y<7;y++){
        for(x=0;x<(6-y);x++)
            printf(" ");
        for(x=6-y;x<7;x++)
            printf("*");
        printf("\n");
    }

    return 0;
}

```

4.

```

#include <stdio.h>
int main(void)

```

```

{
    int x, y, number;

    printf("정수를 입력하시오: ");
    scanf("%d", &number);
    for(y=1;y<=number;y++){
        for(x=1;x<=y;x++){
            printf("%d ", x);
        }
        printf("\n");
    }

    return 0;
}

```

5.

```

#include <stdio.h>
int main(void)
{
    int counter, i;

    printf("카운터의 초기값을 입력하시오: ");
    scanf("%d", &counter);
    for(i=counter;i>=1;i--){
        printf("%d ", i);
    }
    printf("\a");

    return 0;
}

```

6.

// 간단한 산술 계산기 프로그램

```

#include <stdio.h>

int main(void)
{
    char op;
    int x, y;

    while(1){
        printf("*****\n");

```



```

printf("A---- Add \n");
printf("S---- Subtract \n");
printf("M---- Multiply \n");
printf("D---- Divide \n");
printf("Q---- Quit \n");
printf("*****\n");
printf("연산을 선택하시오:");
scanf(" %c", &op);
if( op == 'Q' )
    break;
printf("두수를 공백으로 분리하여 입력하시오: ");
scanf("%d %d", &x, &y);

if( op == 'A' )
    printf("%d \n", x + y);
else if( op == 'S' )
    printf("%d \n", x - y);
else if( op == 'M' )
    printf("%d \n", x * y);
else if( op == 'D' )
    printf("%d \n", x / y);
else
    printf("지원되지 않는 연산자입니다. \n");
}
return 0;
}

```

7.

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

```
int main(void)
```

```
{
```

```
    int i, j;
```

```
    for(i=2; i<=100 ; i++){
```

```
        for(j=2; j<i ; j++){
```

```
            if( i%j == 0 ) goto A;
```

```
        }
```

```
        printf("%d ", i);
```

```
A:
```

```
        ;
```

```
    }
```

```
        return 0;
    }
```

8.

```
#include <stdio.h>
int main(void)
{
    int i, j, height;
    while(1)
    {
        printf("막대의 높이(종료: -1): ");
        scanf("%d", &height);
        if( height == -1 )
            break;
        for(j=0; j<height; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

9. 책의 실행 결과에 오류가 있습니다.

```
#include <stdio.h>
int main(void)
{
    int i, sum;
    i=0;
    sum = 0;
    while(1)
    {
        i++;
        sum += i;
        if( sum >= 10000 )
            break;
    }
    printf("1부터 %d까지의 합이 %d입니다.", (i-1), sum-i);
    return 0;
}
```

1부터 140까지의 합이 9870입니다.

10.

```
#include <stdio.h>
int main(void)
{
    int n, i;
    double r, result;

    printf("실수의 값을 입력하시오: ");
    scanf("%lf", &r);
    printf("거듭제곱횟수를 입력하시오: ");
    scanf("%d", &n);

    result = 1.0;
    for(i=0; i<n; i++)
        result *= r;
    printf("결과값은 %f", result);
    return 0;
}
```

11.

```
#include <stdio.h>
int main(void)
{
    int n, i;
    int result;

    printf("n의 값을 입력하시오: ");
    scanf("%d", &n);

    result = 0;
    for(i=1; i<=n; i++)
        result += i*i;
    printf("계산값은 %d입니다.", result);
    return 0;
}
```

12.

```
#include <stdio.h>
int main(void)
{
```

```

int i, a=0, b=1, c, n;

printf("몇번째 항까지 구할까요? ");
scanf("%d", &n);
for(i=0;i<=n;i++){
    printf("%d, ", a);
    c = a+b;
    a = b;
    b = c;
}
return 0;
}

```

13.

```

#include <stdio.h>
int main(void)
{
    int n, r, result, i;

    printf("n의 값: ");
    scanf("%d", &n);
    printf("r의 값: ");
    scanf("%d", &r);
    result = 1;
    for(i=n;i>=(n-r-1);i--)
        result = result *i;
    printf("%d, ", result);
    return 0;
}

```

14.

```

#include <stdio.h>
#include <math.h>

int main(void)
{
    int x, digit;

    printf("정수를 입력하시오: " );

```

```
scanf("%d", &x);

do {
    digit = x % 10;
    x /= 10;
    printf("%d", digit);
} while(x != 0);

return 0;
}
```

CHAPTER 8

중간점검문제

p.294

1. 가장 중요한 이유는 소스 코드의 중복성을 없애 주기 때문이다.
2. 함수는 프로그램을 이루는 모듈 역할을 한다.
3. 라이브러리 함수

p.297

1. void
2. return (x+y+z);
3. double fabs(double f)

p.307

1. 인수의 값이 매개 변수로 복사된다.
2. 정수가 실수로 형변환된 후에 반환된다.

p.311

1. 반환형, 함수의 이름, 매개 변수--- 함수 헤더라고 한다.
2. 1개
3. void
4. 함수 원형은 컴파일러에게 정보를 주기 위하여 함수 헤더만을 표시한다. 함수 정의에는 반드시 함수 몸체가 있어야 한다.
5. 프로그램의 가독성을 높이기 위하여
6. 함수 pow()는 두 개의 double형 매개 변수를 가지며 반환형은 double형이다.

p.320

1. sin(90.0* (3.141592/180.0));
2. 0에서 9

Summary

함수 원형으로 함수에 대한 정보를 컴파일러에게 제공한다.

- int
- 인수
- 매개 변수
- 사용자 정의, 라이브러리
- { }
- 1개
- void

연습문제

1. (1)
2. (3)
3. (1), (2), (4)
4. (1)
5. (a) 1.720000 (b) 1.000000 (c) 2.000000
- 6.(a) 0에서 9 (b) 2에서 6
7.
 - (a) $y = \log_{10}(x) + \exp(x);$
 - (b) $y = \sin(x) + \sqrt{x^2 - 2.0 \cdot a} + \text{pow}(2.0, 10);$
8.
 - (a) `void print_error(int n);`
 - (b) `double larger_of(double x, double y);`
 - (c) `void side_effect(void);`
9. `int f(void) ----- return 10 + 20;`
`void g(int, int) ----- return;`

```

double h(double, int); ----- return 'a' + 1.0;
10.
#include <stdio.h>
int f(int n);           // 함수 f()의 원형 정의

int main(void)
{
    f(10);              // f()를 인수 10으로 호출
    return 0;
}

int f(int n)
{
    int i, result = 0;

    for(i = 0; i <= n; i++)
        result += i;
    return result;      // 변수 result의 값을 반환
}

```

- 11.
- (a) double f(double x, double y);
 - (b) int f(double x, double y);
 - (c) int f(int x, int y);
 - (d) float get_area(float radius, float pi);

```

7.
#include <stdio.h>
int f(int x);

int f(int x)
{
    int i, sum=0;
    for(i = 0; i<= x; i++)
        sum += i;
    return sum;
}

int main(void)
{
    int n;

```



```

printf("점수를 입력하시오:\n");
scanf("%d", &n);
printf("0부터 %d까지의 합은 %d입니다.\n", n, f(n));
}

```

13.

- (a) `int half_of(int x); -> int half_of(int x)`
- (b) 함수 원형의 매개 변수 개수와 함수 호출시의 인수 개수가 다르다.
- (c) 함수 원형의 매개 변수 타입과 함수 정의 매개 변수 타입이 서로 다르다.
- (d) `sum = x + y -> return x + y;`
- (e) 반환형이 정의되지 않았는데 값을 반환하였다.

프로그래밍 문제

1.

```

#include <stdio.h>

int even(int n);
int absolute(int n);
int sign(int n);

int main(void)
{
    int n;
    printf("정수를 입력하시오:");
    scanf("%d", &n);
    printf("even()의 결과: ");
    if( even(n) == 1 )
        printf("짝수\n");
    else
        printf("홀수\n");
    printf("absolute()의 결과: %d\n", absolute(n));
    printf("sign()의 결과: ");
    if( sign(n) == 1 )
        printf("양수\n");
    else

```

```

        printf("음수\n");
    return 0;
}

```

```

int even(int n)
{
    return (n%2==0)? 1: 0;
}

```

```

int absolute(int n)
{
    return (n>0)? n: -n;
}

```

```

int sign(int n)
{
    return (n>0)? 1: -1;
}

```

2.

```

#include <stdio.h>
double ftoc(double);
int main(void)
{
    double f,c;

    printf("화씨 온도를 입력하시오:");
    scanf("%lf", &f);
    c = ftoc(f);
    printf("섭씨 온도는 %f입니다.\n", c);

    return 0;
}
double ftoc(double f)
{
    return (5.0/9.0)*(f-32.0);
}

```

3.

```

#include <stdio.h>

```

```
double get_radius()
{
    double r;
    printf("원의 반지름을 입력하시오:");
    scanf("%lf", &r);
    return r;
}
double cal_area(double r)
{
    return 3.141592*r*r;
}
```

4.

```
int main(void)
{
    double r, area;
    r = get_radius();
    area = cal_area(r);
    printf("원의 면적은 %f입니다\n", area);
    return 0;
}
#include <stdio.h>
```

```
int get_tax(int income);
```

```
int main(void)
{
    int income;
    printf("소득을 입력하시오(만원):");
    scanf("%d", &income);
    printf("소득세는 %d입니다.\n", get_tax(income));
    return 0;
}
```

```
int get_tax(int income)
{
    if( income > 1000 ) return (int) (income*0.1);
    else return (int) (income*0.08);
}
```

5.

```

#include <stdio.h>
#include <math.h>

double sin_degree(double degree);

int main(void)
{
    double degree;

    for(degree=0.0; degree <= 180.0; degree += 10.0)
        printf("sin(%f)의 값은 %f\n", degree, sin_degree(degree));
    return 0;
}

double sin_degree(double degree)
{
    return sin( (3.141592*degree)/180.0);
}

```

6.

```

#include <stdio.h>
#include <stdlib.h>

int b_rand();
int main(void)
{
    printf("%d\n", b_rand());
    printf("%d\n", b_rand());
    printf("%d\n", b_rand());
    printf("%d\n", b_rand());
    printf("%d\n", b_rand());
    return 0;
}

int b_rand()
{
    return rand()%2;
}

```

7.

```

#include <stdio.h>
#include <stdlib.h>

int b_rand();
int main(void)
{
    int answer, coin;
    char c;

    while(1){
        printf("앞면 또는 뒷면(1 또는 0):");
        scanf("%d", &answer);
        coin = b_rand();
        if( coin == answer )
            printf("맞았습니다.\n");
        else
            printf("틀렸습니다.\n");
        printf("계속하시겠습니까?(y 또는 n):");
        scanf(" %c", &c);
        if( c == 'n' )
            break;
    }
    return 0;
}

int b_rand()
{
    return rand()%2;
}

```

8.

```

#include <stdio.h>
#include <stdlib.h>

double f_rand();
int main(void)
{

    printf("%f\n", f_rand());
    printf("%f\n", f_rand());
    printf("%f\n", f_rand());
}

```

```

        printf("%f\n", f_rand());
        printf("%f\n", f_rand());
        return 0;
    }

double f_rand()
{
    return rand()/(double)RAND_MAX;
}

```

10.

```

#include <stdio.h>
int is_multiple(int n, int m);

int main(void)
{
    int x, y;

    printf("첫번째 정수를 입력하시오:");
    scanf("%d", &x);
    printf("두번째 정수를 입력하시오:");
    scanf("%d", &y);
    if( is_multiple(x, y) == 1 )
        printf("%d는 %d의 배수입니다.\n", x, y);
    else
        printf("%d는 %d의 배수가 아닙니다.\n", x, y);
    return 0;
}

int is_multiple(int n, int m)
{
    if( n%m==0) return 1;
    else return 0;
}

```

11.

7.

```

#include <stdio.h>
#include <math.h>

double get_distance(double x1, double y1, double x2, double y2)

```

```

{
    return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
}

int main(void)
{
    double x1,y1, x2,y2;
    printf("첫번째 점의 좌표를 입력하시오:(x, y)");
    scanf("%lf %lf", &x1, &y1);
    printf("두번째 점의 좌표를 입력하시오:(x, y)");
    scanf("%lf %lf", &x2, &y2);

    printf("두점 사이의 거리는 %f입니다.", get_distance(x1, y1, x2, y2));
    return 0;
}

```

12.

```

#include <stdio.h>
int main(void)
{
    int i, j;

    for(i=2; i<100 ; i++){
        if( is_prime(i) == 1 )
            printf("%d\n", i);
    }

    return 0;
}

int is_prime(int x)
{
    int i;
    for(i=2; i<x ; i++){
        if( x%i == 0 ) return 0;
    }
    return 1;
}

```

13.

```

#include <stdio.h>
#include <math.h>

```

```

long long factorial(int n)
{
    int i;
    long long result = 1;

    for(i = 1; i <= n; i++)
        result *= i;           // result = result * i

    return result;
}

```

```

int main(void)
{
    int i;
    double sum=1.0;
    int n;

    printf("어디까지 계산할까요: ");
    scanf("%d", &n);
    for(i=1;i<=n; i++)
        sum += 1.0/factorial(i);
    printf("오일러의 수는 %f입니다. \n", sum);
    return 0;
}

```

14.

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

```

int is_leap(int y);
int main(void)
{
    int year;

    printf("연도를 입력하시오: ");
    scanf("%d", &year);

    if( is_leap(year) == 1 )
        printf("%d년은 366일입니다.\n", year);
    else
        printf("%d년은 365일입니다.\n", year);
}

```



```

        return 0;
    }

    int is_leap(int y)
    {
        return (y%4==0)&&(y%100!=0) || (y%400);
    }

```

15.

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

```

int round(double a);
int main(void)
{
    double a;

    printf("실수를 입력하시오: ");
    scanf("%lf", &a);

    printf("반올림 한 값은 %d입니다.\n", round(a));

    return 0;
}

int round(double a)
{
    return (int)(a+0.5);
}

```

16.

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

```
#include <math.h>
```

```

int f_equal(double x, double y);
int main(void)
{
    double a, b;

    printf("실수를 입력하시오: ");
    scanf("%lf", &a);
    printf("실수를 입력하시오: ");

```

```

scanf("%lf", &b);

if( f_equal(a, b) == 1 )
    printf("같음\n");
else
    printf("다름\n");

return 0;
}

double f_abs(double x)
{
    if( x > 0 ) return x;
    else return -x;
}
double f_min(double x, double y)
{
    if( x > y ) return y;
    else return x;
}
int f_equal(double x, double y)
{
    double value;
    value = f_abs(x - y)/f_min(f_abs(x), f_abs(y));
    if( value < 0.000001 ) return 1;
    else return 0;
}

```

17.

```

#include <stdio.h>
#include <stdlib.h>

```

```

void print_menu();
void check_menu_number();
void menu_hamburger();
void menu_cheeseberger();
void menu_sandwitch();

```

```

int main(void)
{
    check_menu_number();
    return 0;
}

```

```

}
void menu_hamburger(){
    printf("햄버거 메뉴 호출\n");
}
void menu_cheeseberger(){
    printf("치즈버거 메뉴 호출\n");
}
void menu_sandwitch(){
    printf("샌드위치 메뉴 호출\n");
}

void print_menu()
{
    printf("1. 햄버거\n");
    printf("2. 치즈버거\n");
    printf("3. 샌드위치\n");
    printf("4. 종료\n");
    printf("원하는 메뉴를 선택하시오:");
}

void check_menu_number()
{
    int value;
    while(1){
        print_menu();
        scanf("%d", &value);
        if( value <1 || value > 4 )
            continue;
        else if( value == 1)
            menu_hamburger();
        else if( value == 2)
            menu_cheeseberger();
        else if( value == 3)
            menu_sandwitch();
        else if( value == 4)
            return;
    }
}

```

CHAPTER 9

중간점검문제

p.344

1. 변수가 선언되는 위치
2. 크게 나누어서 지역 변수와 전역 변수가 있다.
3. 전역 변수
4. 지역 변수
5. 없다.
6. 가능하다.
7. 지역 변수는 자동으로 소멸된다.
8. 알 수 없다. 쓰레기값이 들어 있다.
9. 그렇다.
10. 전역 변수는 함수의 외부에 정의될 수 있다.
11. 전역 변수는 프로그램이 실행되는 동안 계속하여서 유지된다. 초기값은 0이다.
12. 지역 변수가 우선시된다.

p.348

1. auto, static, extern, resiter 등이 있다.
2. static
3. register
4. extern
5. 함수(블록)가 종료되어도 소멸되지 않는 변수를 생성하라는 것을 의미한다.

Summary

- 범위, 생존 시간, 연결
- 함수(블록)

- 소멸
- 쓰레기값
- 지역
- 함수
- 0
- 지역
- static
- register
- extern
- 순환
- static

연습문제

1.

```
#include <stdio.h>

void f(void);
double ratio;    // (b)
extern int counter;    // (d)

int main(void)
{
    static int setting;    // (f)
    ...
}

void f(void)
{
    int number; // (a)
    register int index; // (c)
    extern int total; // (e)
    ...
}
```

2.

```
#include <stdio.h>

int a;    // 파일 전체, 정적, 연결 가능
static int b;    // 파일 전체, 정적, 연결 불가능
```

```
extern int c;    // 파일 전체, 정적, 외부 변수 참조
```

```
int main(void)
{
    int d;    // 블록, 자동, 연결 불가능
    register int e;    // 블록, 자동, 연결 불가능
    static int f;    // 블록, 정적, 연결 불가능
    {
        int g;    // 블록, 자동, 연결 불가능
    }
    return 0;
}
```

3.

(a)

// 전역 변수를 사용하여 프로그램이 복잡해지는 경우

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

```
void f(void);
```

```
int i;
```

```
int main(void)
```

```
{
    for(i = 0; i < 3; i++)
    {
        printf("");
        f();
    }
    return 0;
}
```

```
void f(void)
```

```
{
    for(i = 0; i < 5; i++)
        printf("#");
}
```

```
*#####
```

(b)

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

```
void f(int);
```

```
int n = 10;
```

```
int main(void)
```

```
{
    f(n);
    printf("n=%d\n", n);
    return 0;
}
```

```
void f(int n)
{
    n = 20;
}
```

n=10

```
(c)
#include <stdio.h>
void f(void);
int x = 1;

int main(void)
{
    int x = 2;
    printf("%d\n", x);
    {
        int x = 3;
        printf("%d\n", x);
    }
    printf("%d\n", x);
    return 0;
}
```

2
3
2

```
(d)
#include <stdio.h>
void f(void);

int main(void)
{
    f();
    f();
    return 0;
}

void f(void)
{
    static int count = 0;
    printf("%d\n", count++);
}
```

0
1

4.

(a) 하나 이상의 저장 유형 지정자를 붙이면 안된다.

(b) 재귀 호출할 때 매개 변수의 값이 줄어들지 않아서 무한히 재귀 호출됨

5.

(a)

5
4
3
2
1
0

반환값은 16

(b)

5
4
3
2
1
0

반환값은 95

6.

```
int recursive(int n)
{
    int i, sum=0;
    for(i=n; i>=1; i--)
        sum += i;
    return sum;
}
```

프로그래밍 문제

1.

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

```
int add(int, int);
```

```
int sub(int, int);
```

```
int mul(int, int);
```



```
int div(int, int);
```

```
int main(void)
```

```
{
```

```
    char op;
```

```
    int x, y;
```

```
    int i;
```

```
    for(i=0;i<10;i++)
```

```
    {
```

```
        printf("연산을 입력하시오: ");
```

```
        scanf("%d %c %d", &x, &op, &y);
```

```
        if( op == '+' )
```

```
            printf("연산 결과: %d \n", add(x, y));
```

```
        else if( op == '-' )
```

```
            printf("연산 결과: %d \n", sub(x, y));
```

```
        else if( op == '*' )
```

```
            printf("연산 결과: %d \n", mul(x, y));
```

```
        else if( op == '/' )
```

```
            printf("연산 결과: %d \n", div(x, y));
```

```
        else
```

```
            printf("지원되지 않는 연산자입니다. \n");
```

```
    }
```

```
    return 0;
```

```
}
```

```
int add(int x, int y)
```

```
{
```

```
    static int count;
```

```
    count++;
```

```
    printf("덧셈은 총 %d번 실행되었습니다.\n", count);
```

```
    return (x+y);
```

```
}
```

```
int sub(int x, int y)
```

```
{
```

```
    static int count;
```

```
    count++;
```

```
    printf("뺄셈은 총 %d번 실행되었습니다.\n", count);
```

```
    return (x-y);
```

```
}
```

```
int mul(int x, int y)
```

```

{
    static int count;
    count++;
    printf("곱셈은 총 %d번 실행되었습니다.\n", count);
    return (x*y);
}
int div(int x, int y)
{
    static int count;
    count++;
    printf("나눗셈은 총 %d번 실행되었습니다.\n", count);
    return (x/y);
}

```

2.

```

#include <stdio.h>
#include <stdlib.h>

void get_dice_face();

int main(void)
{
    int i;
    for(i=0;i<1000; i++)
        get_dice_face();
    return 0;
}
void get_dice_face()
{
    static int one, two, three, four, five, six;
    int face ;
    face = rand() %6;
    if( face == 0 ) one++;
    else if( face == 1 ) two++;
    else if( face == 2 ) three++;
    else if( face == 3 ) four++;
    else if( face == 4 ) five++;
    else six++;
    printf("%d %d %d %d %d %d\n", one, two, three, four, five, six);
}

```

3.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

```

```

int get_random ()
{
    static int initd=0;
    if( initd == 0 )
    {
        printf("초기화 실행\n");
        srand((unsigned)time(NULL));
        initd = 1;
    }
    return rand();
}

int main(void)
{
    printf("%d\n", get_random());
    printf("%d\n", get_random());
    printf("%d\n", get_random());
    return 0;
}

```

4.

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

```

double recursive(int n)
{
    if( n==1 ) return 1;
    else return 1.0/n+recursive(n-1);
}

```

```

int main(void)
{
    printf("%f\n", recursive(10));
    return 0;
}

```

5.

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

```
#include <stdlib.h>
```

```

void save();
void draw();
int balance;

```

```

int main(void)
{
    int menu;
    while(1){
        printf("메뉴를 선택하세요: 저축(1), 인출(2): ");
        scanf("%d", &menu);
        if( menu == 1 )
            save();
        else if( menu == 2 )
            draw();
        else
            break;
        printf("현재 잔액은 %d입니다.\n", balance);
    }
    return 0;
}

```

```

void save()
{
    int amount;
    printf("저축할 금액: ");
    scanf("%d", &amount);
    balance += amount;
}

```

```

void draw()
{
    int amount;
    printf("인출할 금액: ");
    scanf("%d", &amount);
    balance -= amount;
}

```

6.

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

```

int get_tri_number(int n)
{
    if( n==1 ) return 1;
    else return n+get_tri_number(n-1);
}

```

```

int main(void)
{
    printf("%d\n", get_tri_number(4));
    return 0;
}

```

7.

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

```

int recursive(int n, int k)
{
    if( n==1 || n ==k) return 1;
    else return recursive(n-1, k-1)+ recursive(n-1, k);
}

```

```

int main(void)
{
    printf("%d\n", recursive(3,2));
    return 0;
}

```

8.

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

```

int show_digit(int x)
{
    if( x/10 > 0 )
        show_digit(x/10);
    printf("%d ", x%10);
}

```

```

int main(void)
{
    int n;
    printf("정수를 입력하시오: ");
    scanf("%d", &n);
    show_digit(n);
    return 0;
}

```

CHAPTER 10

중간점검 문제

p.383

1. 관리하기 쉽고 반복이 가능하다.
2. 0
3. n-1
4. 인덱스(첨자)
5. 다른 프로그램의 메모리를 건드릴 수 있다.
6. 사용할 수 없다. 상수만 가능하다.

p.388

1. 배열을 선언하면서 초기화한다면 다음과 같다.

```
int a[6] = { 1, 2, 3, 4, 5, 6 };
```

만약 선언되어 있는 상태이면 다음과 같은 문장으로 초기화가 가능하다.

```
int i;
```

```
for(i=1;i<=6;i++)
```

```
    a[i] = i;
```

2. 초기값의 개수가 적으면 나머지 원소들은 0으로 초기화된다.
초기값의 개수가 많은 경우에는 컴파일 오류가 발생한다.
3. 가능하다.
4. 비교할 수 없다.
5. 대입할 수 없다.

p.400

1. 배열은 원본이 전달된다고 생각할 수 있다.
2. 앞에 `const`를 추가한다.

p.414

1. $3 \times 2 \times 10$ 개의 원소가 존재한다.
2. `int a[3][2][10] = { 0 };`

Summary

- 인덱스
- `a[i]`
- 0, 9
- `int a[6] = { 1, 2, 3, 4, 5, 6 };`
- `int a[] = { 1, 2, 3, 4, 5, 6 };`
- `sizeof(a)/sizeof(a[0])`
- 0
- 30
- 원본
- `const`
- 불가능하다.

연습문제

1. (2)
2. (3)
3. (3), (4)
4. (1)
5. (4)
- 6.

```
#define MAX_SIZE 4
int main(void)
{
    int a[MAX_SIZE] = { 0, 1, 2, 3 };
    int b[4];
    int i;

    for(i=0;i<MAX_SIZE; i++)
        b[i]=a[i];

    return 0;
}
```

```
7.
(a)
float grade[3];
float average, sum;
int i;
for(i=0;i<3;i++){
    printf("성적을 입력하시오: ");
    scanf("%f", &grade[i]);
}
sum = 0.0;
for(i=0;i<3;i++){
    sum += grade[i];
}
average = sum /3.0;
```

(b)


```

float grade[3]={10.0, 20.0, 30.0};
float average, sum;
int i;
sum = 0.0;
for(i=0;i<3;i++){
    sum += grade[i];
}
average = sum /3.0;

```

(c) // 함수 버전

```

float get_average(float a[], int n);
int main(void)
{
    float grade[3]={10.0, 20.0, 30.0};
    get_average(grade, 3);
    return 0;
}

float get_average(float a[], int n)
{
    float average, sum;
    int i;
    sum = 0.0;
    for(i=0;i<3;i++){
        sum += a[i];
    }
    average = sum /3.0;
}

```

(d)

```

void get_freq(float s[], int freq[], int n);
int main(void)
{
    float grade[3] = { 10.0, 20.0, 30.0 };
    int freq[10]={0};
    get_freq(grade, freq, 3);
    return 0;
}

void get_freq(float score[], int f[10], int n)
{
    int i, range;
    for(i=0;i<n;i++){

```

```

        range = score[i]/10;
        f[range]++;
    }

}

```

8.

(a) 2차원 배열을 전달하는 경우에 첫 번째 인덱스의 크기만 생략이 가능하다.

```

int main(void)
{
    int grade[10][60];
    compute_avg(grade);
}

int compute_avg(int array[ ][60]) { ... }

```

(b) 배열 원소인 경우에는 주소를 보내주어야 한다.

```

float test[10];
scanf("%f", &test[0]);

```

(c)

```

int main(void)
{
    int x, y;
    int test[10][5];

    for(x = 0; x < 10; x++)
        for(y = 0; y < 5; y++)
            test[x][y] = 0;

    return 0;
}

```

프로그래밍 문제

1.

```

#include <stdio.h>

int days[] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

int main(void)
{

```

```

    int i;
    for(i=0; i<12; i++)
        printf("%d월은 %d일까지 있습니다.\n", i+1, days[i]);
    return 0;
}

```

2.

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

```

int main(void)
{
    int list[10];
    int i, max, min;

    for(i=0;i<10;i++)
        list[i] = rand();
    max =min=list[0];
    for(i=1;i<10;i++){
        if( list[i] < min )
            min=list[i];
        if( list[i] > max )
            max=list[i];
    }

    printf("최대값은 %d\n", max);
    printf("최소값은 %d\n", min);
    return 0;
}

```

3.

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

```
#define N_DATA 10
```

```
int array_equal(int a[], int b[], int size);
```

```

int main(void)
{
    int A[N_DATA] = { 1, 2, 3 };
    int B[N_DATA] = { 0 };

    printf("%d\n",array_equal(A, B, N_DATA));
}

```

```

        return 0;
    }
    int array_equal(int a[], int b[], int size)
    {
        int i;
        for(i=0; i<size; i++){
            if( b[i] != a[i] )
                return 0;
        }
        return 1;
    }

```

4.

```

#include <stdio.h>
#define N_DATA 10
void array_copy(int a[], int b[], int size);
void array_print(int a[], int size);

int main(void)
{
    int A[N_DATA] = { 1, 2, 3 };
    int B[N_DATA] = { 0 };

    array_copy(A, B, N_DATA);
    array_print(B, N_DATA);
    return 0;
}

void array_copy(int a[], int b[], int size)
{
    int i;
    for(i=0; i<size; i++){
        b[i] = a[i];
    }
}

void array_print(int a[], int size)
{
    int i;
    for(i=0; i<size; i++){
        printf("%d ", a[i]);
    }
    printf("\n");
}

```

```
}
```

5.

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

```
int main(void)
```

```
{
```

```
    int freq[10]={0};
```

```
    int i, max;
```

```
    for(i=0;i<100;i++){
```

```
        ++freq[rand()%10];
```

```
    }
```

```
    max = 0;
```

```
    for(i=1;i<10;i++){
```

```
        if( freq[i] > max )
```

```
            max = i;
```

```
    }
```

```
    printf("%d\n", max);
```

```
    return 0;
```

```
}
```

6.

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

```
int main(void)
```

```
{
```

```
    int i, j, sum;
```

```
    int data[3][5] = { { 12, 56, 32, 16, 98 },
```

```
                        { 99, 56, 34, 41, 3 },
```

```
                        { 65, 3, 87, 78, 21 } };
```

```
    for(i=0; i<3; i++){
```

```
        sum = 0;
```

```
        for(j=0; j<5; j++){
```

```
            sum += data[i][j];
```

```
            printf("%d행의 합계: %d\n", i, sum);
```

```
        }
```

```
    for(j=0; j<5; j++){
```

```
        sum = 0;
```

```
        for(i=0; i<3; i++){
```

```
            sum += data[i][j];
```

```
            printf("%d열의 합계: %d\n", j, sum);
```

```

    }

    return 0;
}
7.
#include <stdio.h>

int main(void)
{
    int list[3][10]={0};
    int value, i;

    for(i=0;i<10;i++){
        list[0][i] = i;
        list[1][i] = i*i;
        list[2][i] = i*i*i;
    }
    printf("정수를 입력하시오:");
    scanf("%d", &value);
    for(i=0;i<10;i++){
        if( value == list[2][i] )
            printf("%d의 세제곱근은 %d\n", value, i);
    }

    return 0;
}

```

```

8.
#include <stdio.h>
#define N_DATA 10
void get_data(double data[]);
double get_mean(double data[]);
double get_std_dev(double data[], double mean);

int main(void)
{
    double data[20];
    double mean;

    get_data(data);
    mean = get_mean(data);
}

```

```

    printf("평균값은 %f\n", mean);
    printf("표준편차값은 %f\n", get_std_dev(data, mean));
    return 0;
}
void get_data(double data[])
{
    int i;
    for(i=0; i<N_DATA; i++){
        printf("데이터를 입력하시오:");
        scanf("%lf", &data[i]);
    }
}
double get_mean(double data[])
{
    int i;
    double sum=0.0;
    for(i=0; i<N_DATA; i++){
        sum += data[i];
    }
    return sum/N_DATA;
}
double get_std_dev(double data[], double mean)
{
    int i;
    double sum=0.0;
    for(i=0; i<N_DATA; i++){
        sum += (data[i]-mean)*(data[i]-mean);
    }
    return sum/N_DATA;
}

```

9.

```

#include <stdio.h>
void get_stat_per_test(int score[][3])
{
    int i, k, min, max, sum;
    for(k=0; k<3; k++){
        min=max=score[0][k];
        sum=0;
        for(i=0; i<10; i++){
            if(score[i][k]<min) min=score[i][k];

```

```

        if(score[i][k]>min) max=score[i][k];
        sum += score[i][k];
    }
    printf("최대 점수=%d\n", max);
    printf("최저 점수=%d\n", min);
}
}
int main(void)
{
    int i, k;
    int score[10][3];
    for(i=0;i<10;i++){
        for(k=0;k<3;k++){
            score[i][k]=rand()%100+1;
        }
    }
    get_stat_per_test(score);
    return 0;
}

```

10.

(a)

```

#include <stdio.h>
void vector_add(double x[], double y[], double z[]);
int main(void)
{
    double x[3]={1.0, 2.0, 3.0};
    double y[3]={1.0, 2.0, 3.0};
    double z[3];

    vector_add(x, y, z);
    return 0;
}
void vector_add(double x[], double y[], double z[])
{
    z[0] =x[0]+y[0];
    z[1] =x[1]+y[1];
    z[2] =x[2]+y[2];
}

```

(b)

```

#include <stdio.h>

```



```

double vector_dotp(double x[], double y[]);
int main(void)
{
    double x[3]={1.0, 2.0, 3.0};
    double y[3]={1.0, 2.0, 3.0};
    double value;

    value = vector_dotp(x, y);
    printf("%f\n", value);
    return 0;
}
double vector_dotp(double x[], double y[])
{
    int i;
    double result;

    result = 0.0;
    for(i=0;i<3;i++)
        result += x[i]*y[i];
    return result;
}

```

11.

```

#include <stdio.h>
int location[] = { 1, 1, 2, 5, 3, 3, 1, 1, 2, 4 };
int main(void)
{
    int product;
    printf("상품 번호를 입력하세요:");
    scanf("%d", &product);
    if( product < 0 || product >= 10 )
        printf("잘못된 상품 번호입니다.\n");
    else
        printf("상품 번호 %d의 위치는 %d입니다.\n", product, location[product]);
    return 0;
}

```

12.

(a) transpose(int a[][3], int b[][3])

```

#include <stdio.h>

```

```

void transpose(int a[][3], int b[][3])
{
    int r,c;
    for(r=0;r<3;r++)
        for(c=0;c<3;c++)
            b[r][c]=a[c][r];
}

void print(int a[][3])
{
    int r,c;
    for(r=0;r<3;r++){
        for(c=0;c<3;c++)
            printf("%d ", a[r][c]);
        printf("\n");
    }
}

int main(void)
{
    int A[3][3]= {{ 1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int B[3][3];

    transpose(A, B);
    print(B);
    return 0;
}

```

```

(b) mult(int a[][3], int b[][3], int c[][3])
#include <stdio.h>
void mult(int A[][3], int B[][3], int C[][3])
{
    int r,c,k;
    for(r=0;r<3;r++){
        for(c=0;c<3;c++){
            int sum = 0;
            for(k=0;k<3;k++)
                sum += A[r][k]*B[k][c];
            C[r][c]=sum;
        }
    }
}

void print(int A[][3])

```

```

{
    int r,c;
    for(r=0;r<3;r++){
        for(c=0;c<3;c++)
            printf("%d ", A[r][c]);
        printf("\n");
    }
}

int main(void)
{
    int A[3][3]= {{ 1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int B[3][3]= {{ 1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int C[3][3];

    mult(A, B, C);
    print(C);
    return 0;
}

```

13.

```

#include <stdio.h>
void show_digit(int n)
{
    int i;
    int binary[32]={0};
    for(i = 0; i < 32 && n > 0; i++)
    {
        binary[i] = n % 2;
        n = n / 2;
    }
    for(i--;i>=0; i--)
    {
        printf("%d", binary[i]);
    }
    printf("\n");
}

int main(void)
{
    show_digit(32);
    return 0;
}

```

14.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
```

```
#define ROWS 20
#define COLS 20
```

```
int grid [ROWS][COLS];
int ro,co;
void mark_grid(int y, int x)
{
    if( x >= 0 && y >= 0 && y < ROWS && x < COLS ){
        ro = y;
        co = x;
        if( grid[y][x] == '.' )
            grid[y][x]= ' *';
    }
}

void print_grid()
{
    int r,c;
    for( r=0; r < ROWS; r++){
        for ( c=0;c< COLS;c++){
            printf("%c ",grid[r][c]);
        }
        printf("\n");
    }
}

int main(){
    int r,c;
    int i;
    char letters = '*';
    for(r=0; r < ROWS; r++){
        for (c=0;c< COLS;c++){
            grid[r][c]='.';
        }
    }

    srand((unsigned)time(NULL));
    ro = ROWS/2;
```

```

co = COLS/2;
grid[ro][co]= letters;

    for(i=0;i<100000; i++){
        int move = rand() % 8;
        switch (move){
            case 0:    mark_grid(ro+1, co); break;
            case 1:    mark_grid(ro+1, co+1); break;
            case 2:    mark_grid(ro, co+1); break;
            case 3:    mark_grid(ro-1, co+1); break;
            case 4:    mark_grid(ro-1, co); break;
            case 5:    mark_grid(ro-1, co-1); break;
            case 6:    mark_grid(ro, co-1); break;
            case 7:    mark_grid(ro+1, co-1); break;
        }
        print_grid();
        if( getch() == 'q' ) break;
    }

    return 0;
}

```

15.

```

#include <stdio.h>
#include <stdlib.h>
void sieve(int s[], int size);

int main(void)
{
    int    top_value = 100;
    int    count     = top_value - 1;
    int    my_array[101];
    int    i, prime, multiple;
    for (i=2; i <= top_value; ++i)
        my_array[i] = 1;
    for (prime = 2; prime <= top_value; ++prime)
    {
        if (my_array[prime])
            for (multiple = 2*prime; multiple <= top_value; multiple +=
prime)
                my_array[multiple] = 0;
    }
}

```

```
}  
for (i=2; i <= top_value; ++i)  
{  
    if (my_array[i])  
        printf("%d ", i);  
}  
return 0;  
}
```

CHAPTER 11

중간점검문제

p,435

1. 바이트(byte)
2. (a) 1 (b) 2 (c) 4 (d) 4 (e) 4 (f) 8
3. 포인터도 변수의 일종이다. 따라서 저장하고 있는 값의 변경이 가능하다.
4. & 연산자
5. `p = &x;`
6. `*p = 25;`

p,443

1. * 연산자, ++, --, +, - 연산자
2. 84 번지를 가리킨다.
3. `*p++` : p가 가리키는 값을 사용한 후에 p를 증가
`(*p)++` : p가 가리키는 값을 증가
4. `*(p+3)`은 p가 가리키는 위치로부터 3번째에 있는 객체

p,447

1. (1) A (2) `&A[0]`
2. 첫 번째 원소의 값
3. 대입할 수 없다. 배열의 이름은 포인터 상수라고 생각하면 된다. 즉 변경이 불가능하다.
4. 참조할 수 있다.
5. 사용할 수 있다. `p[0]`, `p[1]` ,... 등의 표현이 가능하다.

p,457

1. 값에 의한 호출(call by value)
2. 참조에 의한 호출(call by reference)
3. 배열은 복사되지 않고 원본이 전달된다.

Summary

- 바이트
- 주소
- `&x`
- `*p = 100;`
- `NULL`
- 첫 번째 원소
- 2
- `p`가 가리키는 위치에서 값을 추출한 후에 `p`를 증가시키는 것
- `+`, `-`
- 84
- 값

연습문제

1.
 - (a) `*(list+6)`
 - (b) `*(name+3)`
 - (c) `cost[8]`
 - (d) `*(message+0)`

2.
`char *p;`
`p = &code;`
`*p = 'a';`

3. (3)
4. (3)
- 5.

5
5
계속하려면 아무 키나 누르십시오 . . .

6.

1008
2008
계속하려면 아무 키나 누르십시오 . . .

7.

0
1
1
계속하려면 아무 키나 누르십시오 . . .

8.

(a) void print_array(double a[])

(b) void print_array(double *a)

9.

ip의 값이 전달되었기 때문에 ip를 변경할 수 없다. ip를 변경하려면 ip의 주소를 전달하여야 한다.

프로그래밍 문제

1.

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

```
int main(void) {
```

```
    int x = 0x12345678;
```

```
    unsigned char *xp = (char *)&x;
```

```
    printf("바이트순서: %x %x %x %x\n", xp[0], xp[1], xp[2], xp[3]);
```

```
    return 0;
```

```
}
```

바이트순서: 78 56 34 12

계속하려면 아무 키나 누르십시오 . . .

인텔 CPU는 리틀 엔디안임을 알 수 있다.

2.

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

```
void get_sum_diff(int x, int y, int *p_sum, int *p_diff);
```

```
int main(void)
```

```
{
```

```
    int sum=0, diff=0;
```

```
    get_sum_diff(100, 200, &sum, &diff);
```

```
    printf("원소들의 합=%d\n", sum);
```

```
    printf("원소들의 차=%d\n", diff);
```

```
    return 0;
```

```
}
```

```
void get_sum_diff(int x, int y, int *p_sum, int *p_diff)
```

```
{
```

```
    *p_sum = x+y;
```

```
    *p_diff = x-y;
```

```
}
```

3.

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

```
#include <stdlib.h>
```

```
void array_fill(int *A, int size);
```

```
void array_print(int *A, int size);
```

```
int main(void)
```

```
{
```

```
    int data[10];
```

```
    array_fill(data, 10);
```

```
    array_print(data, 10);
```

```
    return 0;
```

```
}
```

```
void array_fill(int *A, int size)
```

```
{
```

```
    int i;
```

```

        for(i=0;i<size; i++){
            A[i] = rand();
        }
    }
void array_print(int *A, int size)
{
    int i;
    for(i=0;i<size; i++){
        printf("%d ", A[i]);
    }
    printf("\n");
}

```

4.

```

#include <stdio.h>

void array_print(int *A, int n);
int main(void)
{
    int list[10]={ 1, 2, 3, 4 };

    array_print(list, 10);
    return 0;
}
void array_print(int *A, int n)
{
    int i;
    printf("A[]={ ");
    for(i=0;i<n;i++){
        printf("%d ", A[i]);
    }
    printf(" }\n");
}

```

5.

```

#include <stdio.h>
void convert(double *grades, double *scores, int size);
void print(double *a, int size)

```

```

{
    int i;
    for(i=0; i<size; i++){
        printf("%f ", a[i]);
    }
    printf("\n");
}

int main(void)
{
    double grades[10] = { 1, 2, 3, 4, 3, 2,1, 2,3, 4 };
    double scores[10] = {0};
    convert(grades, scores, 10);
    print(scores, 10);
    return 0;
}

void convert(double *grades, double *scores, int size) {
    int i;
    for(i=0 ; i<size ; i++) {
        scores[i] = 100.0*(grades[i]/4.3);
    }
}

```

6.

```

#include <stdio.h>
#define N_DATA 10
void array_copy(int *a, int *b, int size);
void array_print(int *a, int size);

int main(void)
{
    int A[N_DATA] = { 1, 2, 3 };
    int B[N_DATA] = { 0 };

    array_copy(A, B, N_DATA);
    array_print(B, N_DATA);
    return 0;
}

void array_copy(int *a, int *b, int size)
{
    int i;

```

```

        for(i=0; i<size; i++){
            b[i] = a[i];
        }
    }
void array_print(int *a, int size)
{
    int i;
    for(i=0; i<size; i++){
        printf("%d ", a[i]);
    }
    printf("\n");
}

```

7.

```

#include <stdio.h>
#define N_DATA 10
void array_add(int a[], int b[], int c[], int size);

```

```

int main(void)
{
    int A[N_DATA] = { 1, 2, 3 };
    int B[N_DATA] = { 0 };
    int C[N_DATA] = { 0 };

    array_add(A,B,C, N_DATA);
    return 0;
}
void array_add(int a[], int b[], int c[], int size)
{
    int i;
    for(i=0; i<size; i++){
        c[i]=a[i]+b[i];
    }
}

```

8.

```

#include <stdio.h>
#define N_DATA 10
int array_sum(int a[], int size);

```

```

int main(void)

```

```

{
    int A[N_DATA] = { 1, 2, 3 };
    int sum ;
    sum = array_sum(A, N_DATA);
    printf("%d \n", sum);
    return 0;
}
int array_sum(int a[], int size)
{
    int i;
    int sum=0;
    for(i=0; i<size; i++){
        sum += a[i];
    }
    return sum;
}

```

9.

```

#include <stdio.h>
int search(int *A, int size, int x);

int main(void)
{
    int data[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int index = search(data, 10, 9);
    printf("%d\n", index);
    return 0;
}
int search(int *A, int size, int x)
{
    int i;
    for(i=0; i<size; i++){
        if( A[i] == x ) return i;
    }
    return -1;
}

```

10.

```

#include <stdio.h>

```

```

void get_gcd_lcm(int x, int y, int *gcd, int *lcm);
int main(void)
{
    int x, y, g, l;
    printf("두개의 정수를 입력하시오: ");
    scanf("%d %d", &x, &y);
    get_gcd_lcm(x, y, &g, &l);
    printf("최소공배수는 %d입니다.\n", l);
    printf("최대공약수는 %d입니다.\n", g);
    return 0;
}

void get_gcd_lcm(int org_x, int org_y, int *gcd, int *lcm)
{
    int x = org_x;
    int y = org_y;
    int tmp;

    while(y != 0)
    {
        tmp = y;
        y = x % y;
        x = tmp;
    }
    *gcd = x;
    *lcm = org_x * org_y / *gcd;
}

```

11.

/* i는 정렬된 왼쪽리스트에 대한 인덱스
j는 정렬된 오른쪽리스트에 대한 인덱스
k는 정렬될 리스트에 대한 인덱스 */

```

void merge(int left, int mid, int right)
{
    int i, j, k, l;
    i=left; j=mid+1; k=left;

    /* 분할 정렬된 list의 합병 */
    while(i<=mid && j<=right)
    {
        if(list[i]<=list[j])
            sorted[k++] = list[i++];
    }
}

```

```

        else
            sorted[k++] = list[j++];
    }
    if(i>mid)    /* 오른쪽리스트에 남아 있는 레코드의 일괄 복사 */
        for(l=j; l<=right; l++)
            sorted[k++] = list[l];
    else /* 왼쪽리스트에 남아 있는 레코드의 일괄 복사 */
        for(l=i; l<=mid; l++)
            sorted[k++] = list[l];
    /* 배열 sorted[]의 리스트를 배열 list[]로 재복사 */
    for(l=left; l<=right; l++)
        list[l] = sorted[l];
}

```


CHAPTER 12

중간점검문제

p.478

1. 문자 배열로 표현된다.
2. 문자열의 끝을 표현한다.
3. 0
4. 시스템이 문자열의 끝을 알 수 없어서 이상한 글자들이 출력된다.
5. B는 변수의 이름으로 생각한다. 'B'는 하나의 문자이다. "B"는 문자 B로만 이루어진 문자열을 의미한다.
6. 메모리의 데이터 세그먼트에 저장된다.
7. NULL 문자를 저장하기 위하여
8. (a) `char s[6] = "Hello";` (b) `char s[6] = { 'H', 'e', 'l', 'l', 'o', '\0' };`

p.481

1. `getchar()`는 버퍼를 사용하므로 엔터키가 눌러져야만이 입력을 가져간다. 반면에 `_getch()`는 버퍼를 사용하지 않고 글자가 입력되는 대로 전달한다.
2. (1) `getchar()` 또는 `getch()` 사용
(2) `scanf("%c", &ch);`

p.483

1.
`char line[100];`
`gets(line);`
2.
`char word[100];`
`scanf("%s", word);`

p.487

1. `ctype.h`
2. `getchar()`는 버퍼를 사용하므로 엔터키가 눌러져야만이 입력을 가져간다. 반면에

_getch()는 버퍼를 사용하지 않고 글자가 입력되는 대로 전달한다.

3. 16
4. 'A'

p,495

1. strcpy(s2, s1);
2. 크기가 7이상인 char 배열 필요
3. strcmp()
4. strncpy()에서는 복사하는 문자열의 크기를 제한할 수 있다.
5. strcat(s2, s2);
6. 0

p,498

1. 실수 3.141592가 차지하는 공간은 8바이트가 된다. (double형 상수)
반면에 "3.141592"가 차지하는 공간은 9바이트가 된다. (NULL 문자 포함)
2. atof(), 또는 sscanf()
3. s가 붙으면 문자열에 출력한다.

p,502

1.
 - (1) char s[5][10] = { "C", "JAVA", "C++", "BASIC" };
 - (2) char* s2[] = { "C", "JAVA", "C++", "BASIC" };
2. printf("%s\n", s[0]);

Summary

- 아스키
- char형
- NULL
- 문자

대상	설명
A	변수로 간주
'A'	하나의 문자
"A"	문자열

- char s[] = "xyz";
- NULL

함수	설명
strlen(s)	문자열 s의 길이
strcpy(s1, s2)	s2를 s1에 복사한다.
strcat(s1, s2)	s2를 s1의 끝에 붙여넣는다.
strcmp(s1, s2)	s1과 s2를 비교한다. 일치하면 0

- 0
- string.h
- 버퍼를 사용하느냐 사용하지 않느냐
- 문자열의 크기를 제한하느냐 제한하지 않느냐
- 0
- printf()는 콘솔로 출력한다. sprintf()는 문자열에 출력한다.

함수	설명
gets(s)	한 줄의 문자열을 s로 입력받는다.
puts(s)	문자열 s를 콘솔에 출력한다.
sprintf(s, ...)	문자열 s에 출력한다.
scanf(s, ...)	문자열 s로부터 읽는다.

연습문제

1.

- (a) strcat()
- (b) strcpy()
- (c) strtok()
- (d) gets()
- (e) strlen()

2.

- (a) '?'을 “?”로 변경
- (b) if(strcmp(s, "value")==0)
- (c) strcpy(a, "Hello World!");

3. s1이 가리키는 것은 문자열 상수로서 더 이상의 추가 공간을 가지고 있지 않기 때문에 s2가 가리키는 문자열을 저장할 수 없다.
수정한 결과는 다음과 같다.

```
char s1[20] = "Hi! ";  
char *s2 = "Programmers";  
strcat(s1, s2);
```

4. 첫번째 문장에서는 문자 배열이 선언되고 문자 배열의 초기값이 "Hello World!"가 된다. 두 번째 문장에서는 읽기 전용 메모리에 문자열 상수가 저장되고 이 문자열 상수의 주소가 포인터 p에 대입된다.

5. (4)

6. 문자형 포인터 p가 초기화가 되어 있지 않다. 따라서 p가 가리키는 곳에 문자열을 저장할 수 없다. 올바르게 수정하면 다음과 같다.

```
int main(void)
```

```
{  
    char p[100];  
    scanf("%s", p);  
}
```

7. (a) 4 바이트 (b) 하나의 포인터가 생성되므로 4 바이트
(c) 10 바이트 (d) 20바이트

8. (a) "HIGH" (b) 'D' (c) "HIGH"

9.

```
o  
lo  
llo  
ello  
Hello  
계속하려면 아무 키나 누르십시오 . . .
```

10.

mystery()는 s1과 s2가 같으면 0을 반환하고 다르면 1을 반환한다. mystery("abc", "abd")와 같이 호출하면 1이 반환된다.

프로그래밍 문제

1.

```
#include <stdio.h>

int main(void)
{
    char ch;
    printf("문자를 입력하시오: ");
    scanf("%c", &ch);
    printf("아스키 코드값=%d\n", ch);
    return 0;
}
```

2.

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

#define SIZE 100

void delete_space(char s[])
{
    char tmp[SIZE];
    int i, k=0;
    for(i=0;i<(int)strlen(s);i++){
        if( s[i] != ' ' )
            tmp[k++] = s[i];
    }
    tmp[k]=0;
    strcpy(s, tmp);
}

int main(void)
{
    char str[SIZE];
    printf("공백 문자가 있는 문자열을 입력하시오: ");
    gets(str);
}
```

```

        delete_space(str);
        printf("%s", str);
        return 0;
}

```

3.

```
#include <string.h>
```

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

```
#define SIZE 100
```

```
int str_chr(char *s, int c)
```

```

{
    int i;
    int count=0;
    for(i=0;i<strlen(s);i++){
        if( s[i] == c )
            count++;
    }
    return count;
}

```

```
int main(void)
```

```

{
    char str[SIZE];
    char ch;

    printf("문자열을 입력하시오: ");
    gets(str);

    printf("개수를 셀 문자를 입력하시오: ");
    ch = getchar();
    printf("%c의 개수: %d", ch, str_chr(str, ch));
    return 0;
}

```

4.

```
#include <string.h>
```

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

```
#define SIZE 100
```

```
int str_chr(char *s, int c)
{
    int i;
    int count=0;
    for(i=0;i<strlen(s);i++){
        if( s[i] == c )
            count++;
    }
    return count;
}
```

```
int main(void)
{
    char str[SIZE];
    char ch;

    printf("문자열을 입력하시오: ");
    gets(str);

    for(ch='a';ch<='z';ch++){
        printf("%c: %d \n", ch, str_chr(str, ch));
    }
    return 0;
}
```

5.

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

```
int main(void)
{
    char c;
    while(1){
        printf("문자를 입력하시오: ");
        c=getchar();
        if( islower(c) )
            putchar(toupper(c));
        if( isupper(c) )
```

```

        putchar(tolower(c));
    if( !isalpha(c) )
        printf("경고 ");
    if( c=='.' ) break;
    fflush(stdin); // 줄바꿈 문자 제거
}
return 0;
}

```

6.

```

#include <string.h>
#include <stdio.h>

```

```

#define SIZE 100

```

```

void str_upper(char *s)
{
    int i;
    int count=0;
    for(i=0;i<strlen(s);i++){
        if( s[i] >='a' && s[i] <= 'z' )
            s[i] = s[i]-'a'+'A';
    }
}

```

```

int main(void)
{
    char str[SIZE];
    printf("문자열을 입력하시오: ");
    gets(str);

    str_upper(str);
    printf("변환된 문자열: %s\n", str);
    return 0;
}

```

7.

```

#include <stdio.h>
#include <string.h>

```



```

int get_response(char *prompt)
{
    char response[100];
    printf(prompt);
    scanf("%s", response);
    if( strcmp(response, "yes") == 0 ||
        strcmp(response, "y") == 0 ||
        strcmp(response, "YES") == 0 ||
        strcmp(response, "Y") == 0 )
        return 1;
    else return 0;

}

int main(void)
{
    int result;

    result = get_response("게임을 하시겠습니까");
    if( result == 1 )
        printf("긍정적인 답변);
    else
        printf("부정적인 답변);
    return 0;
}

```

8.

```
#include <string.h>
```

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

```
char s[] = "Man is immortal, because he has a soul";
```

```
char seps[] = " ,\t\n";
```

```
char *token;
```

```
int main( void )
```

```
{
```

```
    int count=0;
```

```
    token = strtok( s, seps );
```

```

while( token != NULL )
{
    count++;
    token = strtok( NULL, seps );
}
printf("단어의 수는 %d입니다.\n", count);
return 0;
}

```

9.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>

```

```

int main(void)
{
    char s[200]={0};
    printf("텍스트를 입력하시오:");
    gets(s);
    if( !isupper(s[0]) )
        s[0] = toupper(s[0]);
    if( s[strlen(s)-1] != '.' ){
        s[strlen(s)] = '.';
        s[strlen(s)+1] = NULL;
    }
    printf("결과 텍스트 출력:%s\n", s);
    return 0;
}

```

10.

```

#include <stdio.h>
#include <string.h>

```

```

int main(void)
{
    char s[100];
    int i, len;

    printf("문자열 입력");
    scanf("%s", s);
    len = strlen(s);

```

```

        for(i=0;i<len/2;i++)
            if( s[i] != s[len-i-1] ){
                printf("회문이 아님");

            }

        printf("회문임");
        return 0;
    }

```

11. 포인터 배열도 이용하여야 한다.

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

```
#include <string.h>
```

```
int main(void)
```

```
{
```

```
    char s[100];
```

```
    char *ptr[100];
```

```
    int i=0;
```

```
    int j;
```

```
    puts("문자열을 입력하시오: ");
```

```
    gets(s);
```

```
    ptr[0]=strtok(s, " ");
```

```
    while(ptr[i] != NULL)
```

```
    {
```

```
        ptr[i+1]=strtok(NULL, " ");
```

```
        i++;
```

```
    }
```

```
    for(j=i-1;j>=0;j--)
```

```
    {
```

```
        printf("출력 문자열: %s ",ptr[j]);
```

```
    }
```

```
    return 0;
```

```
}
```

12.

```

#include <string.h>
#include <stdio.h>
#define SIZE 100

int main(void)
{
    char seps[] = ", ";
    char s[SIZE];
    char *token, *name, *fname;
    int i;

    printf("성과 이름을 대문자로 입력하시오: ");
    gets(s);
    token = strtok( s, seps ); // 문자열에서 첫번째 토큰을 얻는다.
    fname = token;
    token = strtok( NULL, seps ); // 다음 토큰을 얻는다.
    name = token;

    for(i=0;i<strlen(name);i++)
        name[i] = tolower(name[i]);
    for(i=0;i<strlen(fname);i++)
        fname[i] = tolower(fname[i]);
    printf("%s, %s\n", name, fname);
    return 0;
}

```

13.

```

#include <string.h>
#include <stdio.h>

#define SIZE 100

int get_punc(char *s)
{
    int i;
    int count=0;
    for(i=0;i<strlen(s);i++){
        if( s[i] == ',' || s[i] == '.' )
            count++;
    }
    return count;
}

```

```
}
```

```
int main(void)
```

```
{
```

```
    char str[SIZE];
```

```
    printf("문자열을 입력하시오: ");
```

```
    gets(str);
```

```
    printf("구두점의 개수는 %d입니다.\n", get_punc(str));
```

```
    return 0;
```

```
}
```

14.

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

```
#include <string.h>
```

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
    char seps[] = " ";
```

```
    char s[200], find[100], replace[100], target[200]="";
```

```
    char *token;
```

```
    printf("문자열을 입력하시오: ");
```

```
    gets(s);
```

```
    printf("찾을 문자열: ");
```

```
    gets(find);
```

```
    printf("바꿀 문자열: ");
```

```
    gets(replace);
```

```
    token = strtok( s, seps ); // 문자열에서 첫번째 토큰을 얻는다.
```

```
    while( token != NULL )
```

```
    {
```

```
        if( strcmp(token, find)==0 )
```

```
            strcat(target, replace);
```

```
        else
```

```
            strcat(target, token);
```

```
        token = strtok( NULL, seps ); // 다음 토큰을 얻는다.
```

```
        strcat(target, " ");
```

```
    }
```

```
    printf("결과: %s", target);
```

```
    return 0;
```

```
}
```

15.

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

```
#include <string.h>
```

```
int main(void)
```

```
{
```

```
    char s[100]={0};
```

```
    char op[100];
```

```
    int x,y;
```

```
    char *token;
```

```
    printf("연산을 입력하십시오:");
```

```
    gets(s);
```

```
    token = strtok( s, " " ); // 문자열에서 첫번째 토큰을 얻는다.
```

```
    strcpy(op, token);
```

```
    token = strtok( NULL, " " ); // 다음 토큰을 얻는다.
```

```
    x =atoi(token);
```

```
    token = strtok( NULL, " " ); // 다음 토큰을 얻는다.
```

```
    y =atoi(token);
```

```
    if( strcmp(op, "add")==0 ){
```

```
        printf("연산의 결과: %d", x+y);
```

```
    }
```

```
    else if( strcmp(op, "sub")==0 ){
```

```
        printf("연산의 결과: %d", x-y);
```

```
    }
```

```
    else if( strcmp(op, "mul")==0 ){
```

```
        printf("연산의 결과: %d", x*y);
```

```
    }
```

```
    else if( strcmp(op, "div")==0 ){
```

```
        printf("연산의 결과: %d", x/y);
```

```
    }
```

```
    else {
```

```
    }
```

```
    return 0;
```

```
}
```

16.

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

int main(void)
{
    char s[100];
    int size, i, k;

    puts("광고하고 싶은 텍스트를 입력하시오: ");
    gets(s);
    size = strlen(s);
    puts("=====");
    for(k=0;k<size; k++){
        for(i=k;i<(k+size); i++){
            if( i < size)
                putchar(s[i]);
            else {
                putchar(s[i-size]);
            }
        }
        putchar('\n');
    }
    return 0;
}
```

CHAPTER 13

중간점검문제

p.517

1. 구조체는 서로 타입이 다른 변수들을 묶는 것이다. 배열은 타입이 같은 변수들을 묶는 것이다.

2. // x값과 y값으로 이루어지는 화면의 좌표

```
struct point {  
    int x;           // x 좌표  
    int y;           // y 좌표  
};
```

// 복소수

```
struct complex {  
    double real;     // 실수부  
    double imag;     // 허수부  
};
```

// 날짜

```
struct date {  
    int month;  
    int day;  
    int year;  
};
```

// 사각형

```
struct rect {  
    int x;  
    int y;  
    int width;  
    int height;  
};
```


p,528

1. = 연산자
2. 구조체 태그는 구조체의 정의를 나타낸다. 따라서 구조체 태그에는 메모리 공간이 할당되지 않는다. 반면에 구조체 변수에는 실제로 메모리 공간이 할당된다.
3. 가능하다.
4. 가능하다.

p,531

```
1.  
struct product {  
    int number;  
    char name[100];  
    int price;  
};  
struct product parray[5];
```

p,541

1. 구조체는 복사본이 전달된다.
2. 매개 변수 선언시에 **const**를 붙인다.

p,546

1. union
2. 가장 큰 멤버의 크기와 같다.

p,549

1. enum
2. 어떤 변수가 정해진 값만을 갖는 경우에 선언한다.
3. 0부터 시작되는 정수값이 할당된다.

p,553

1. 사용자가 새로운 타입을 선언할 수 있게 한다.
2. 이식성을 높이고 문서화의 역할도 하며 컴파일러가 점검할 수 있도록 한다.
- 3.

```
// 직원  
typedef struct {  
    char name[20]; // 이름  
    int age;        // 나이  
    int gender;     // 성별
```

```
        int salary;        // 월급
    } employee;
```

Summary

- 구조체는 서로 다른 타입의 변수들을 묶는 것이고 배열은 같은 타입의 변수들을 묶은 것이다.
- **struct, union, enum**
- 만들어지지 않는다. 구조체 변수를 선언하여야 한다.
- **p->x**
- 구조체 매개 변수 앞에 **const**를 추가한다.
- **union**
- **enum**
- 특정한 숫자 대신에 기호를 사용함으로써 프로그램의 이해도를 향상시킬 수 있고 변수가 열거된 값 이외의 값을 취하는 것을 막아 에러를 줄여준다.
- **typedef**

Exercise

1.

```
struct customer {
    char name[20]; // 이름
    int zip_code; // 주소
    long mileage; // 마일리지 정보
};
struct customer c1;
```

2.

- (a) 구조체를 선언하면 자동으로 변수가 생성된다. -> 거짓
- (b) **typedef**은 변수를 선언하는 키워드이다. -> 거짓
- (c) 구조체는 == 연산자를 사용하여 비교할 수 있다. -> 거짓
- (d) 구조체를 함수로 전달하면 원본이 전달된다. -> 거짓
- (e) 구조체 변수는 =연산자를 이용하여 대입될 수 있다. -> 참

3. (2)

4. (1), (3)

5.

`enum colors { white, red=3, blue, green, black=9 };`

식별자	white	red	blue	green	black
값	0	3	4	5	9

6. (1), (3), (6), (7), (8)

7. 공용체에서는 모든 멤버를 동시에 초기화할 수 없다.

8.

(a)

```
struct book{
    char author[30];
    int pub_date;
    int pages;
    int price;
};
```

(b)

```
struct friend{
    char name[30];
    int age;
    double height;
};
```

(c)

```
struct part{
    char name[10];
    int num;
    int price;
};
```

9.

(a)

`enum primary_color { RED, GREEN, BLUE };`

(b)

`enum months { Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec };`

10.

(a)

```
struct book{
    char title[50];
    int pages;
```

```
};
```

```
book.pages = 512;
```

→book.pages 가 아니라 구조체 변수명.pages 가 되어야 옳다.

(b)

```
struct book {
```

```
    char title[50] = "Data Structures in C";
```

```
    int pages = 577;
```

```
} abook;
```

→

```
struct book {
```

```
    char title[50];
```

```
    int pages;
```

```
} abook = { "Data Structures in C", 577};
```

(c)

```
typedef enum { red, green, blue } color;
```

```
color.red = 1;
```

->

```
typedef enum { red, green, blue } color;
```

```
color c;
```

```
c=red;
```

(d)

```
struct fraction {
```

```
    int num;
```

```
    int den;
```

```
} *p;
```

```
*p.num = 3;
```

```
*p.den = 5;
```

→

컴파일 오류, p는 구조체를 가리키는 포인터로 선언된다. p가 초기화가 안되어 있다.

프로그래밍 문제

1.

(a)

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

```
struct point {
```

```

        int x, y;
};
int equal(struct point p1, struct point p2);
int main(void)
{
    struct point p1={1, 2};
    struct point p2={3, 5};
    printf("일치 여부 = %d\n", equal(p1, p2));
    return 0;
}
int equal(struct point p1, struct point p2)
{
    if( p1.x == p2.x && p1.y == p2.y ) return 1;
    else return 0;
}

```

(b)

```

#include <stdio.h>
struct point {
    int x, y;
};
int quadrant(struct point p);
int main(void)
{
    struct point p={-1, 2};
    printf("사분면 = %d\n", quadrant(p));
    return 0;
}
int quadrant(struct point p)
{
    if( p.x > 0 && p.y > 0 ) return 1;
    else if( p.x < 0 && p.y > 0 ) return 2;
    else if( p.x < 0 && p.y < 0 ) return 3;
    else return 4;
}

```

2.

(a)

```

#include <stdio.h>
struct point {
    int x, y;
};

```

```

int equal(struct point *p1, struct point *p2);
int main(void)
{
    struct point p1={1, 2};
    struct point p2={3, 5};
    printf("일치 여부 = %d\n", equal(&p1, &p2));
    return 0;
}
int equal(struct point *p1, struct point *p2)
{
    if( p1->x == p2->x && p1->y == p2->y ) return 1;
    else return 0;
}

```

(b)

```

#include <stdio.h>
struct point {
    int x, y;
};
int quadrant(struct point *p);
int main(void)
{
    struct point p={-1, 2};
    printf("사분면 = %d\n", quadrant(&p));
    return 0;
}
int quadrant(struct point *p)
{
    if( p->x > 0 && p->y > 0 ) return 1;
    else if( p->x < 0 && p->y > 0 ) return 2;
    else if( p->x < 0 && p->y < 0 ) return 3;
    else return 4;
}

```

3.

```

#include <stdio.h>
#include <math.h>
struct point {
    int x, y;
};
struct rectangle {
    struct point a, b;          // a는 오른쪽 상단, b는 왼쪽 하단을 나타낸다.
}

```

```

};
int area(struct rectangle r);
int perimeter(struct rectangle r);
int is_square(struct rectangle r);
int main(void)
{
    struct point p1={1, 1};
    struct point p2={2, 2};
    struct rectangle r;
    r.a = p1;
    r.b = p2;

    printf("%d\n", area(r));
    printf("%d\n", perimeter(r));
    printf("%d\n", is_square(r));
    return 0;
}
int area(struct rectangle r)
{
    return abs(r.a.x-r.b.x)*abs(r.a.y-r.b.y);
}
int perimeter(struct rectangle r)
{
    return 2*(abs(r.a.x-r.b.x)+abs(r.a.y-r.b.y));
}
int is_square(struct rectangle r)
{
    return abs(r.a.x-r.b.x)==abs(r.a.y-r.b.y);
}

```

4.

```

#include <stdio.h>
struct complex {
    double real;
    double imag;
};

struct complex add(struct complex c1, struct complex c2)
{
    struct complex result;
    result.real = c1.real+c2.real;

```

```

        result.imag = c1.imag+c2.imag;
        return result;
    }
    struct complex sub(struct complex c1, struct complex c2)
    {
        struct complex result;
        result.real = c1.real+c2.real;
        result.imag = c1.imag+c2.imag;
        return result;
    }
    struct complex mul(struct complex c1, struct complex c2)
    {
        struct complex result;
        result.real = c1.real*c2.real-c1.imag*c2.imag;
        result.imag = c1.real*c2.imag+c1.imag*c2.real;
        return result;
    }
    void print(struct complex c)
    {
        printf("%f+%fi\n", c.real,c.imag);
    }

    int main(void)
    {
        struct complex c1={ 1.0, 2.0};
        struct complex c2={ 2.0, 3.0};
        struct complex c3;

        c3= add(c1, c2);
        print(c3);

        return 1;
    }

```

5.

```

#include <stdio.h>
#include <math.h>
struct vector {
    double x;
    double y;
};

```



```

struct vector vector_add(struct vector v1, struct vector v2)
{
    struct vector r;
    r.x = v1.x + v2.x;
    r.y = v1.y + v2.y;
    return r;
}
void vector_print(struct vector v)
{
    printf("(%f, %f)\n", v.x, v.y);
}
int main(void)
{
    struct vector v1={ 1.0, 2.0 };
    struct vector v2={ 2.0, 3.0 };
    struct vector v3;
    v3 = vector_add(v1, v2);
    vector_print(v3);
    return 0;
}

```

6.

```

#include <stdio.h>
#include <string.h>
struct email {
    char title[100];
    char receiver[50];
    char sender[50];
    char content[1000];
    char date[100];
    int priority;
};
int main(void)
{
    struct email e;
    strcpy(e.title, "안부 메일");
    strcpy(e.receiver, "chulsoo@hankuk.ac.kr");
    strcpy(e.sender, "hsh@hankuk.ac.kr");
    strcpy(e.content, "안녕하십니까? 별일 없으신지요?");
    strcpy(e.date, "2010/9/1");
    e.priority = 1;
}

```

```

        return 0;
}

```

7.

```

#include <stdio.h>
#include <math.h>
struct food {
    char name[100];
    int calories;
};
int calc_total_calroies(struct food array[], int size);
int main(void)
{
    struct food food_array[3]=
        {"hamburger", 900}, {"bulgogi", 500}, {"sushi", 700}};
    int total = calc_total_calroies(food_array, 3);
    printf("%d\n", total);
    return 0;
}
int calc_total_calroies(struct food array[], int size)
{
    int i, total=0;
    for(i=0;i<size;i++){
        total += array[i].calories;
    }
    return total;
}

```

8.

```

#include <stdio.h>
#include <string.h>
// 직원
struct employee {
    int number;    // 사번
    char name[20]; // 이름
    int age;       // 나이
    char tel[20];  // 전화번호
};
int main(void)
{
    struct employee e[10] = {

```

```

        { 1, "홍길동1", 20, "111-1111" },
        { 2, "홍길동2", 25, "111-1112" },
        { 3, "홍길동3", 60, "111-1113" },
        { 4, "홍길동4", 40, "111-1114" },
        { 5, "홍길동5", 50, "111-1115" },
        { 6, "홍길동6", 45, "111-1116" },
        { 7, "홍길동7", 32, "111-1117" },
        { 8, "홍길동8", 23, "111-1118" },
        { 9, "홍길동9", 29, "111-1119" },
        { 10, "홍길동10", 62, "111-1120" } };

int i;
for(i=0;i<10;i++)
    if( e[i].age >= 20 && e[i].age <= 30 )
        printf("%s\n", e[i].name);

return 0;
}

```

9.

```

#include <stdio.h>
#include <math.h>
struct contact {
    char name[100];
    char home_phone[100];
    char cell_phone[100];
};

int main(void)
{
    struct contact list[5];
    int i;
    char name[100];
    for(i=0;i<5;i++){
        printf("이름을 입력하시오:");
        scanf("%s", list[i].name);
        printf("집전화번호를 입력하시오:");
        scanf("%s", list[i].home_phone);
        printf("휴대폰번호를 입력하시오:");
        scanf("%s", list[i].cell_phone);
    }
    printf("검색할 이름을 입력하시오:");
    scanf("%s", name);
    for(i=0;i<5;i++){
        if( strcmp(name, list[i].name)==0 ){

```

```

        printf("집 전화번호: %s\n", list[i].home_phone);
        printf("휴대폰 번호: %s\n", list[i].cell_phone);
        return 0;
    }
}
printf("검색이 실패하였슴\n");

return 0;
}

```

10.

```

#include <stdio.h>
#include <math.h>
struct card {
    int value;
    char suit;
};
int main(void)
{
    struct card cards[52];
    int i;
    for(i=0;i<13;i++){
        cards[i].value=i%13+1;
        cards[i].suit='c';
    }
    for(i=0;i<13;i++){
        cards[i+13].value=i%13+1;
        cards[i+13].suit='d';
    }
    for(i=0;i<13;i++){
        cards[i+26].value=i%13+1;
        cards[i+26].suit='h';
    }
    for(i=0;i<13;i++){
        cards[i+39].value=i%13+1;
        cards[i+39].suit='s';
    }
    for(i=0;i<52;i++){
        printf("%d:%c ", cards[i].value, cards[i].suit);
    }
    return 0;
}

```

15.

```
#include <stdio.h>
#include <math.h>
enum shape_type { TRIANGLE, RECTANGLE, CIRCLE };
struct shape {
    int type;
    union {
        struct {
            int base, height;
        } tri;
        struct {
            int width, height;
        } rect;
        struct {
            int radius;
        } circ;
    } p;
};

int main(void)
{
    struct shape s;
    enum shpae_type type;
    printf("도형의 타입을 입력하시오(0, 1, 2): ");
    scanf("%d", &type);
    switch(type){
    case TRIANGLE:
        printf("밑변과 반지름을 입력하시오(예를 들어서 100 200): ");
        scanf("%d %d", &s.p.tri.base, &s.p.tri.height);
        printf("면적은 %d\n", (int)(0.5*s.p.tri.base*s.p.tri.height));
        break;
    case RECTANGLE:
        printf("가로와 세로의 길이를 입력하시오(예를 들어서 100 200):");
        scanf("%d %d", &s.p.rect.width, &s.p.rect.height);
        printf("면적은 %d\n", (int)(s.p.rect.width*s.p.rect.height));
        break;
    case CIRCLE:
        printf("반지름을 입력하시오(예를 들어서 100): ");
        scanf("%d", &s.p.circ.radius);
        printf("면적은 %d\n", (int)(3.14*s.p.circ.radius*s.p.circ.radius));
```

```

        break;
    }
    return 0;
}
18.
#include <stdio.h>
#include <string.h>
#define TITLE_SIZE 50
#define NAME_SIZE 50
#define LOCATION_SIZE 50

enum music_type{KPOP, POP, CLASSIC, SCREEN_MUSIC};
typedef struct music{
    char title[TITLE_SIZE];
    char singer[NAME_SIZE];
    char location[LOCATION_SIZE];
    enum music_type genre;
} MUSIC;

void add_record(MUSIC library[], int count);
void menu();
int get_input();
void search_record(MUSIC library[], int count);
void print_record(MUSIC library[], int count);
int main(void)
{
    int num, count = 0;
    MUSIC library[30] = {"\0"};
    while(1)
    {
        menu();
        num = get_input();
        switch(num)
        {
            case 1:
                add_record(library, count);
                count++;
                continue;
            case 2:
                print_record(library, count);
                continue;

```

```

        case 3:
            search_record(library, count);
            continue;
        case 4:
            return -1;
        }
        return 0;
    }
}

void add_record(MUSIC library[], int count)
{
    int type;

    fflush(stdin);
    printf("제목:");
    gets(library[count].title);
    printf("가수:");
    gets(library[count].singer);
    printf("위치:");
    gets(library[count].location);
    printf("장르(0: 가요, 1: 팝, 2: 클래식, 3: 영화음악)");
    scanf("%d",&type);
    if(type >= KPOP && type <= SCREEN_MUSIC)
        library[count].genre = type;
    else
        library[count].genre = KPOP;
}

void menu()
{
    printf("=====\n");
    printf(" 1. 추가\n");
    printf(" 2. 출력\n");
    printf(" 3. 검색\n");
    printf(" 4. 종료\n");
    printf("=====\n");
}

int get_input()
{
    int num;
    printf("정수값을 입력하시오 : ");

```

```

        scanf("%d",&num);
        return num;
    }
void search_record(MUSIC library[], int count)
{
    int i;
    char title[TITLE_SIZE];

    fflush(stdin);
    printf("제목: ");
    gets(title);

    for(i = 0; i < count; i++)
    {
        if(strcmp(title,library[i].title) == 0)
        {
            printf("저장된 위치는 %s\n",library[i].location);
            return;
        }
    }
    printf("찾는 음악이 테이블에 없습니다.\n");
}
void print_record(MUSIC library[], int count)
{
    int i;
    fflush(stdin);
    for(i = 0; i < count; i++)
    {
        printf("제목 : %s\n",library[i].title);
        printf("가수 : %s\n",library[i].singer);
        printf("위치 : %s\n",library[i].location);
        if(library[i].genre == 0)
            printf("장르 : 가요\n");
        else if(library[i].genre == 1)
            printf("장르 : 팝\n");
        else if(library[i].genre == 2)
            printf("장르 : 클래식\n");
        else if(library[i].genre == 3)
            printf("장르 : 영화음악\n");
    }
}

```


CHAPTER 14

중간점검문제

p.569

1. `double **dp;`
2. `**dp`는 변수 `c`를 가리킨다.

p.572

1. `double *p[10];`
2. 2차원 배열에서는 사용하지 않는 공간이 낭비될 수 있다. 래그드 배열에서는 낭비되는 공간이 없다.

p.581

1. `int (*pf)(double);`
2. `pf(3.0);`

p.586

1. `m[0]`은 2차원 배열 `m`에서 0번째 행을 가리킨다.
2. `(m+1)`은 1번째 행을 가리킨다.

p.588

1.
`void *vp;`
`int *ip;`
`ip = (int *)vp;`

p.592

1. `main`
2. 4

Summary

// 설명을 빈칸에 적으면서 정리합니다.

```
...
int main(int argc, char *argv[])
{
    int x, y;
    int *p[2] = { &x, &y };
    int (*pf)(int, int);

    x = atoi(argv[1]);
    y = atoi(argv[2]);
    check(p);

    pf = add;
    printf("%d\n", pf(x, y));
    return 0;
}

void check(int *p[2])
{
    if( *p[0] < 0 )      *p[0] = 0;
    if( *p[1] < 0 )      *p[1] = 0;
}

int add(int x, int y)
{
    return x+y;
}
```

외부로부터 인수를 받을 수 있는 main() 함수

포인터의 배열이다.

함수 포인터이다.

함수 포인터에 함수의 주소를 대입

함수 포인터를 통하여 함수를 호출

- * 이중 포인터
- * 포인터 배열
- * 함수 포인터
- * 1번째 행
- * void 포인터
- * 개수, 명령어의 인수

연습문제

1. 다음은 무엇을 선언하는 문장인가?

(a) `int **dp;`

`int`형 이중 포인터 변수 `dp` 선언

(b) `int *pa[10];`

`int`형 포인터 배열 `pa` 선언

(c) `int (*p)[10];`

`int`형 배열 포인터 `p` 선언

(d) `void (*f)(int a);`

`int`형 매개 변수를 가지고 반환형이 없는 함수를 가리키는 포인터 `f` 선언

4. 다음의 설명에 맞는 문장을 작성하라.

(a) `int **dp;`

(b) `double *pa[5];`

(c) `double (*pf)(char);`

(d) `int (*pf[10])(short, short);`

(e) `void *p;`

5. 다음 프로그램의 출력을 쓰시오

(a)

```
int i = 80;
```

```
int *p = &i;
```

```
int **dp = &p;
```

```
**dp = 90;
```

```
*p = 100;
```

```
printf("%d %d %d", i, *p, **dp);
```

→100 100 100

(b)

```
int m[3][2] = {1, 2, 3, 4, 5, 6};
```

```
printf("%d\n", **m);
```

```
printf("%d\n", *m[1]);
```

→

1

3

(c)

```
char *a[] = { "서울", "부산", "인천", "대구"};
```

```
char **x;
```

```
x = a;  
printf("%s\n", *x);
```

→

서울

(e)

```
int a[2][3] = { {1, 2, 3}, {4, 5, 6} };  
int (*p)[3] = a;
```

```
printf("%d", p[0][1]);
```

→

2

프로그래밍 문제

1.

// 이중 포인터 프로그램

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

```
void set_proverb(char **q, int n);
```

```
int main(void)
```

```
{
```

```
    char *s = NULL;
```

```
    int n;
```

```
    printf("몇번째 속담을 선택하시겠습니까?");
```

```
    scanf("%d", &n);
```

```
    set_proverb(&s, n);
```

```
    printf("selected proverb = %s\n",s);
```

```
    return 0;
```

```
}
```

```
void set_proverb(char **q, int n)
```

```
{
```

```
    static char *array[10]={"A bad shearer never had a good sickle.",
```

```

        "A bad workman (always) blames his tools. ",
        "A bad workman quarrels with his tools.",
        "A bad thing never dies.",
        "A bad workman finds fault with his tools.",
        "A bad workman always blames his tools.",
        "A bargain is a bargain.",
        "A barking dog never bites.",
        "A barking dog was never a good hunter."};
*q = array[n];
}

```

2.

```

#include <stdio.h>

int get_sum(int array[], int size);
int main(void)
{
    int m[][1]= { 1, 1, 1 };

    int result = get_sum(m[0], 1);
    printf("%d\n", result);

    return 0;
}

int get_sum(int array[], int size)
{
    int i=0;
    int sum=0;
    for(i=0;i<size; i++)
        sum += array[i];
    return sum;
}

```

3. 문제 오타 수정해주세요! 함수의 원형이 `pr_str_array(char **dp, int n)`이어야 합니다.

```

#include <stdio.h>

void pr_str_array(char **dp, int n)
{
    int i;
    for(i=0;i<n;i++){

```

```

        printf("%s\n", dp[i]);
    }
}

int main(void)
{
    char *array[10]={"A bad shearer never had a good sickle.",
        "A bad workman (always) blames his tools. ",
        "A bad workman quarrels with his tools.",
        "A bad thing never dies.",
        "A bad workman finds fault with his tools.",
        "A bad workman always blames his tools.",
        "A bargain is a bargain.",
        "A barking dog never bites.",
        "A barking dog was never a good hunter."};
    pr_str_array(array, 10);
    return 0;
}

```

4.

```

#include <stdio.h>
void set_max_ptr(int m[], int size, int **pmax);
int main(void)
{
    int m[6]={ 5,6,1,3,7,9};
    int *pmax;
    set_max_ptr(m, 6, &pmax);
    printf("가장 큰 값은 %d\n", *pmax);
    return 0;
}

void set_max_ptr(int m[], int size, int **pmax)
{
    int i=0;
    int max;
    max = m[0];
    for(i=1;i<6;i++){
        if( m[i] > max ){
            max = m[i];
            (*pmax) = &m[i];
        }
    }
}

```

```
}
```

5.

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

```
void sort_strings(char *s[], int size);
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    char *s[3] = {"mycopy", "src", "dst"};
```

```
    sort_strings(s, 3);
```

```
    for(i=0;i<3;i++)
```

```
        printf("%s\n", s[i]);
```

```
    return 0;
```

```
}
```

```
void sort_strings(char *list[], int size)
```

```
{
```

```
    int i, j, least;
```

```
    char *temp;
```

```
    for(i = 0; i < size-1; i++)
```

```
    {
```

```
        least = i;    // i번째 값을 최소값으로 가정
```

```
        for(j = i + 1; j < size; j++)    // 최소값 탐색
```

```
            if(strcmp(list[j], list[least]) < 0)
```

```
                least = j;
```

```
        // i번째 원소와 least 위치의 원소를 교환
```

```
        temp = list[i];
```

```
        list[i] = list[least];
```

```
        list[least] = temp;
```

```
    }
```

```
}
```

6.

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

```
int get_average(int list[], int n);
```

```
int main(void)
```

```
{
```

```
    int score[3][3]={100, 30, 67},
```

```
    {89, 50, 12},
```

```
    {19, 60, 90}};
```

```

    int i, avg;
    for(i=0;i<3;i++){
        avg = get_average(score[i], 3);
        printf("%d행의 평균값=%d\n", i, avg);
    }
    return 0;
}

int get_average(int list[], int n)
{
    int i;
    int sum = 0;
    for(i=0;i<n;i++)
        sum += list[i];
    return sum/n;
}

```

7.

(a) 이진화

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

```
#define WIDTH 10
```

```
#define HEIGHT 10
```

```
void threshold_image(int p[HEIGHT][WIDTH], int h, int w);
```

```
int main(void)
```

```

{
    int image[HEIGHT][WIDTH] = {
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,0,9,9,0,0,0,0,0},
        {0,0,9,0,9,0,0,0,0,0},
        {0,0,0,0,8,0,0,0,0,0},
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,0,0,7,0,0,0,0,0},
        {0,0,0,0,8,0,0,0,0,0},
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,6,6,9,7,7,0,0,0}};
    brighten_image(image, HEIGHT, WIDTH);

    return 0;
}

```

```
void threshold_image(int p[HEIGHT][WIDTH], int h, int w)
```



```

{
    int x, y;
    for(y = 0; y < h; y++)
        for(x = 0; x < w; x++)
            if(p[y][x] > 5) p[y][x] = 255;
            else p[y][x] = 0;
}

```

(b) 반전

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

```
#define WIDTH 10
```

```
#define HEIGHT 10
```

```
void reverse_image(int p[HEIGHT][WIDTH], int h, int w);
```

```
int main(void)
```

```

{
    int image[HEIGHT][WIDTH] = {
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,0,9,9,0,0,0,0,0},
        {0,0,9,0,9,0,0,0,0,0},
        {0,0,0,0,8,0,0,0,0,0},
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,0,0,7,0,0,0,0,0},
        {0,0,0,0,8,0,0,0,0,0},
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,0,0,9,0,0,0,0,0},
        {0,0,6,6,9,7,7,0,0,0}};
    reverse_image(image, HEIGHT, WIDTH);

    return 0;
}

```

```
void reverse_image(int p[HEIGHT][WIDTH], int h, int w)
```

```

{
    int x, y;
    for(y = 0; y < h; y++)
        for(x = 0; x < w; x++)
            p[y][x] = 255 - p[y][x];
}

```

8.

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

```
#define HEIGHT 10
```

```
#define WIDTH 10
```

```
void array_copy(int src[][WIDTH], int dst[][WIDTH]);
void array_print(int a[][WIDTH]);
```

```
int score[HEIGHT][WIDTH]={100, 30, 67},
                        {89, 50, 12},
                        {19, 60, 90}};
int tmp[HEIGHT][WIDTH];
```

```
int main(void)
{
    array_copy(score, tmp);
    array_print(tmp);
    return 0;
}

void array_print(int a[][WIDTH])
{
    int r,c;
    for(r=0;r<HEIGHT;r++){
        for(c=0;c<WIDTH;c++)
            printf("%d ", a[r][c]);
        printf("\n");
    }
}
```

```
void array_copy(int src[][WIDTH], int dst[][WIDTH])
{
    int *p=&src[0][0];
    int *end=&src[HEIGHT-1][WIDTH-1];
    int *q=&dst[0][0];
    while(p <= end){
        *q++ = *p++;
    }
}
```

9.

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

```
#define HEIGHT 10
```

```
#define WIDTH 10
```

```
#define ALIVE 1
```

```
#define DEAD 0
```

```
void print_board(int board[HEIGHT][WIDTH]) {  
    int h, w;  
    for (h = 0; h < HEIGHT; h++) {  
        for (w = 0; w < WIDTH; w++) {  
            if (board[h][w] == ALIVE) {  
                printf("X");  
            } else {  
                printf("-");  
            }  
        }  
        printf("\n");  
    }  
    printf("\n");  
}
```

```
void clear_board(int board[HEIGHT][WIDTH]) {  
    int h, w;  
    for (h = 0; h < HEIGHT; h++) {  
        for (w = 0; w < WIDTH; w++) {  
            board[h][w] = DEAD;  
        }  
    }  
}
```

```
int get_value(int board[HEIGHT][WIDTH], int row, int col) {  
    if (row < 0 || row >= HEIGHT  
        || col < 0 || col >= WIDTH  
        || board[row][col] != ALIVE )  
    {  
        return 0;  
    }  
    else {  
        return 1;  
    }  
}
```

```

int get_neighbor_count(int board[HEIGHT][WIDTH], int row, int col) {

    int neighbor = 0;
    if( row == 0 || col == 0 || row ==(HEIGHT-1) || col ==(WIDTH-1)){
        // 가장자리이면 표준적인 방법
        neighbor += get_value(board, row - 1, col - 1);
        neighbor += get_value(board, row - 1, col);
        neighbor += get_value(board, row - 1, col + 1);
        neighbor += get_value(board, row, col - 1);
        neighbor += get_value(board, row, col + 1);
        neighbor += get_value(board, row + 1, col - 1);
        neighbor += get_value(board, row + 1, col);
        neighbor += get_value(board, row + 1, col + 1);
    }
    else {
        // 내부이면 빠른 방법
        int *p = &(board[row][col]);
        neighbor += *(p-WIDTH-1);
        neighbor += *(p-WIDTH);
        neighbor += *(p-WIDTH+1);
        neighbor += *(p-1);
        neighbor += *(p+1);
        neighbor += *(p+WIDTH-1);
        neighbor += *(p+WIDTH);
        neighbor += *(p+WIDTH+1);
    }
    return neighbor;
}

```

```

void do_generation(int board[HEIGHT][WIDTH]) {

    int tmp_board[HEIGHT][WIDTH];
    int neighbor, h, w;
    for (h = 0; h < HEIGHT; h++) {
        for (w = 0; w < WIDTH; w++) {
            neighbor = get_neighbor_count(board, h, w);
            if (neighbor==3) {
                tmp_board[h][w] = ALIVE;
            } else if (neighbor == 2 && board[h][w] == ALIVE) {

```

```

        tmp_board[h][w] = ALIVE;
    } else if (neighbor >= 4 && board[h][w] == ALIVE) {
        tmp_board[h][w] = DEAD;
    } else {
        tmp_board[h][w] = DEAD;
    }
}
}

for (h = 0; h < HEIGHT; h++) {
    for (w = 0; w < WIDTH; w++) {
        board[h][w] = tmp_board[h][w];
    }
}
}

```

```

void initialize_board(int board[HEIGHT][WIDTH])
{
    board[3][4] = ALIVE;
    board[3][5] = ALIVE;
    board[3][6] = ALIVE;
    board[4][6] = ALIVE;
    board[5][5] = ALIVE;
}

```

```

int main(void)
{
    int board[HEIGHT][WIDTH];

    char user;
    int generation = 0;

    clear_board(board);
    initialize_board(board);
    print_board(board);
    do {
        do_generation(board);
    }
}

```

```
    print_board(board);
    printf("generation %d\n", ++generation);
    printf("종료하려면 q, 계속하려면 m: ");
    scanf(" %c", &user);
} while (user != 'q') ;
return 0;
}
```

CHAPTER 15

중간점검문제

p.606

1. #define KEY 1234
2. #define INPUT scanf

p.614

1. 함수 매크로가 함수 호출 오버헤드가 없어서 빠르게 실행된다.
2. #define CUBING(x) ((x)*(x)*(x))

p.618

1.

```
#ifdef TEST
    printf("TEST\n");
#endif
```

p.621

1.

```
#if DEBUG==2
    printf("DEBUG\n");
#endif
```
2.

```
#if (DEBUG==2) && (LEVEL == 3)
    printf("DEBUG\n");
#endif
```

p.631

1. 여러 소스 파일로 만드는 편이 소스를 유지, 보수, 관리하기가 편리하다.
2.

```
// factorial.h
```

```

#ifndef FACTORIAL_H
#define FACTORIAL_H

int factorial(int x);

#endif

// factorial.c
#include "factorial.h"

int factorial(int x)
{
    int result = 1;
    int i;

    for(i = x; i >= 1; i--)
        result *= i;

    return result;
}

```

3.

```

// point.h
#ifndef POINT_H
#define POINT_H

struct point {
    int x;
    int y;
};

#endif

```

p.633

1. 비트 필드 구조체
2. unsigned 형이나 unsigned int형

Summary

- define
- 함수 매크로

```
#define PI      3.141592      // 단수 매크로 정의
#define CUBE(x) ((x)*(x)*(x)) // 함수 매크로 정의 , 주어진 값을 세제곱한다.
```

- #ifdef

```
#define CPLUSPLUS 1
#ifdef CPLUSPLUS      // CPLUSPLUS가 정의되어 있으면
#define NULL 0
#else                  // CPLUSPLUS가 정의되어 있지 않으면
#define NULL (void *)0
#endif
```

- #if

```
#define CPLUSPLUS 1
#if CPLUSPLUS==1      // CPLUSPLUS의 값이 1이면
#define NULL 0
#else
#define NULL (void *)0
#endif
```

- #undef
- 줄번호
- 현재의 소스 파일 이름
- extern

```
typedef struct Foo {    // 비트 필드 구조체
    int flag    : 1;
    int counter : 15;
} Foo;
```

```
Foo myFoo;
```

```
myFoo.flag = 1;
```

- 비트 필드 구조체
- int 또는 unsigned int

연습문제

1.

```
#if VERSION==1 //단순 매크로 VERSION이 1이면
    printf("원의 면적은 %f입니다\n", area(radius));
#elif VERSION==2 //단순 매크로 VERSION이 2이면
    printf("area of the circle is %f \n", area(radius));
#else //단순 매크로 VERSION이 1이나 2가 아니면
    printf("area of the circle is %f \n", area(radius));
#endif
```

2. (4)

3. (1) 오류 : 함수 매크로의 매개 변수 y를 사용하지 않는다. -> 매크로를 실제로 호출할 때 오류가 발생한다.

(2) 오류: #if VERSION==1 으로 변경하여야 한다.

(3) 오류: #if (DEBUG > 3.00)으로 변경
전처리기에서는 실수를 사용할 수 없다.

(4) 오류가 없다.

(5) 오류: #define SIZE 100으로 변경

(6) 오류: #if (VERSION*100 > 300 && DATE > 80901)으로 변경, 정수 앞에 0이 붙으면 오류 발생

(7) 오류: 문자열과 비교할 수 없음!

4. #include <header>은 컴파일러가 설치된 디렉토리에서 헤더 파일을 찾는다. #include "header"은 현재 디렉토리에서 헤더 파일을 찾는다.

5. #define AREA(w, h) w*h -> #define AREA(w, h) ((w)*(h))

6.

DEBUG 버전

계속하려면 아무 키나 누르십시오 . . .

7.

(a)

firstsecond

(b)

Kim and Park Hello!

8.

(a)

```
#ifdef !DEBUG
```

```
->
```

```
#ifndef DEBUG
```

(b)

```
#define PRINT(x) printf("%d", x);
```

```
int main(void)
```

```
{
```

```
    PRINT(x);
```

```
    return 0;
```

```
}
```

```
->
```

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

```
#define PRINT(x) printf("%d", x);
```

```
int main(void)
```

```
{
```

```
    int x=0;
```

```
    PRINT(x);
```

```
    return 0;
```

```
}
```

9.

(a) struct test {

```
    unsigned bit1:1;
```

```
    unsigned bit2:1;
```

```
    unsigned bit3:1;
```

```
    unsigned bit4:1;
```

```
    unsigned bit5:1;
```

```
    unsigned bit6:1;
```

```
    unsigned bit7:1;
```

```
    unsigned bit8:1;
```

```
};
```

(b)

```
struct test x;
```

```
x.bit1=1;
```

프로그래밍 문제

1.

```
double power(int x, int y)
{
    double result = 1.0;
    int i;

    for(i = 0; i < y; i++)
    {
        printf("result=%f\n", result);    //①
        result *= x;
    }

    return result;
}

(a)
#ifdef DEBUG
    printf("result=%f\n", result);    //①
#endif

(b)
#if DEBUG==2
    printf("result=%f\n", result);    //①
#endif

(c)
#if (DEBUG==2 && LEVEL==3)
    printf("result=%f\n", result);    //①
#endif

(d)
printf("%d: result=%f\n", __LINE__, result); //①

(e)
#if POWER_TYPE==0
int power(int x, int y)
{
    int result = 1.0;
#else
double power(int x, int y)
{
    double result = 1.0;
#endif
...
#if POWER_TYPE==0
    printf("result=%d\n", result);    //①
#else
    printf("result=%f\n", result);    //①
```

```

#endif
...
(f)
#if 0
                printf("result=%f\n", result);        //①
#else

```

2.

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

```

#define MIN(x,y) (x)<(y)?(x):(y)
#define GET_MIN(x,y,z) (MIN(x, y)<(z)?(MIN(x, y)):(z))
int main(void)
{
    int x, y, z, result;
    printf("3개의 정수를 입력하시오: ");
    scanf("%d %d %d", &x, &y, &z);
    result = GET_MIN(x, y, z);
    printf("최소값은 %d입니다. ", result);
    return 0;
}

```

3.

```

#define ARRAY_INIT(array, size, value)    { int i; for(i=0;i<size;i++)\
                                           array[i]=(value); }

int main(void)
{
    int a[10];
    ARRAY_INIT(a, 10, 0);
    return 0;
}

```

4.

```

#include <stdio.h>
#define VOLUME(r, h) (3.141592*r*r*h)

int main(void)
{
    double r, h, volume;
    printf("원기둥의 반지름을 입력하시오: ");
    scanf("%lf", &r);
    printf("원기둥의 높이를 입력하시오: ");
    scanf("%lf", &h);

```

```

        volume = VOLUME(r, h);
        printf("원기둥의 부피: %f ", volume);
        return 0;
    }

```

5.

```

#include <stdio.h>
#define IS_SPACE(c) ((c)==' ' || (c)=='\t' || (c)=='\n')

```

```

int main(void)
{
    char str[100];
    int i, count;

    printf("문자열을 입력하시오: ");
    gets(str);

    i=0;
    count=0;
    while( str[i] != NULL ) {
        if( IS_SPACE(str[i]) )
            count++;

        i++;
    }
    printf("공백문자의 개수: %d \n", count);
    return 0;
}

```

6.

```

#include <stdio.h>
#define SET_BIT(n, pos) ((n) |= (1 << (pos)))
#define GET_BIT(n, pos) ((n) & (1 << (pos)))
#define CLR_BIT(n, pos) ((n) &= ~(1 << (pos)))

```

```

int main(void)
{
    int n;
    n = 0x808081;
    printf("GET_BIT(): %d \n", GET_BIT(n, 0));
    SET_BIT(n, 2);
    printf("%#x \n", n);
}

```

```

        return 0;
    }

```

7. (힌트)에 오차가 있습니다.

```

for(i=0;i<32;i++)
    if( GET_BIT(n, i) )
        printf("1");
    else
        printf("0");
->
for(i=31;i>=0;i--)
    if( GET_BIT(n, i) )
        printf("1");
    else
        printf("0");

```

```

#include <stdio.h>

```

```

#define GET_BIT(n, pos) ((n) & (1 << (pos)))

```

```

void display_bit(int value)

```

```

{
    int i;
    for(i=31;i>=0;i--)
        if( GET_BIT(value, i) )
            printf("1");
        else
            printf("0");
    printf("\n");
}

```

```

int main(void)

```

```

{
    int n, orientation, distance;
    printf("정수값을 입력하시오: ");
    scanf("%d", &n);
    display_bit(n);
    return 0;
}

```

8.

```

#include <stdio.h>

```

```

#define GET_BIT(n, pos) ((n) & (1 << (pos)))

```

```

void display_bit(int value)
{
    int i;
    for(i=31;i>=0;i--)
        if( GET_BIT(value, i) )
            printf("1");
        else
            printf("0");
    printf("\n");
}

int main(void)
{
    int n, orientation, distance;
    printf("정수값을 입력하시오: ");
    scanf("%d", &n);
    printf("왼쪽 이동은 0, 오른쪽 이동은 1을 입력하시오: ");
    scanf("%d", &orientation);
    printf("이동시킬 거리: ");
    scanf("%d", &distance);

    printf("이동 전: ");
    display_bit(n) ;
    printf("이동 후: ");
    if( orientation == 0 )
        display_bit(n << distance);
    else
        display_bit(n >> distance);
}

```

9.

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

```

int main(void)
{
    char s[100];
    int i;

    printf("문자열을 입력하시오: ");
    scanf("%s", s);

    for(i=0;i<strlen(s);i++){
        s[i] = s[i] ^ 0x20;
    }
}

```



```

    }
    printf("결과 문자열: %s\n", s);
    return 0;
}

```

10.

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

```
void encode(char *src, char *key, char *result);
```

```
void decode(char *src, char *key, char *result);
```

```
void main(void)
```

```

{
    char *key = "12345678";
    char *data = "I am a boy";
    char result1[100];
    char result2[100];

    encode(data, key, result1);
    printf("%s가 %s로 인코딩됨\n", data, result1);

    decode(result1, key, result2);
    printf("%s가 %s로 디코딩됨\n", result1, result2);
}

```

```
void encode(char *src, char *key, char *result)
```

```

{
    while(*src) *result++ = *src++ ^ *key++;
    *result = '\0';
}

```

```
void decode(char *src, char *key, char *result)
```

```

{
    encode(src, key, result);
}

```

I am a boy가 xRY1W|Zoy로 인코딩됨
xRY1W|Zoy가 I am a boy로 디코딩됨
계속하려면 아무 키나 누르십시오 . . .

CHAPTER 16

중간점검문제

p.646

1. 스트림
2. 바이트
3. 장치 독립성
4. stdin, 키보드
5. stdout, 콘솔 화면

p.650

1. 형식 제어 문자열
2. 오른쪽
3. %d, %i
4. %o, %d, %x
5. %e
6. 6
7. 10.6
8. -
9. 6

p.655

1. %lf
2. 공백 문자

p.662

1. 바이트
2. 텍스트, 이진
3. fopen()
4. FILE

p.669

1. int
2. fgets()
3. fprintf()
4. fscanf()

p.677

1. 이진
2. fopen()
3. "rb"

p.681

1. 순차접근
2. 임의접근
3. 파일 위치 표시자
4. ftell()

Summary

```
#include <stdio.h>
int main(void)
{
    FILE *fpt, *fpb;
    int i = 1234567890;

    fpt = fopen("test.txt", "w");
    fpb = fopen("test.dat", "wb");

    if( fpt == NULL || fpb == NULL )
    {
        printf("파일을 오픈할 수 없습니다.\n");
        return 1;
    }
    fprintf(fpt, "%d", i);
    fwrite(&i, sizeof(int), 1, fpb);

    fclose(fpt);
    fclose(fpb);

    return 0;
}
```

스트림을 나타내는 FILE
구조체를 가리키는 포인터

파일은 fopen()을 호출하
여서 연다.

fopen()의 반환값이 NULL
이면 파일이 제대로 열리
지 않은 것이다.

형식을 지정하여서 파일에
데이터를 기록한다.

이진 데이터 형식으로 파
일에 데이터를 기록한다.

파일을 닫는다.

}

- * 바이트
- * FILE
- * stdin, stdout, stderr
- * 텍스트, 이진
- * EOF
- * "rb+" 또는 "wb+"
- * int
- * remove()
- * 임의 접근
- * ftell()

연습문제

1. (4)
2. (1) fflush()
(2) fgetc()
(3) feof()
(4) fscanf()

3. (3)
- 4.

(a)

```
int i;  
scanf("%d", i);
```

->

```
int i;  
scanf("%d", &i);
```

(b)

```
double d;  
scanf("%f", &d);
```

->

```
double d;  
scanf("%lf", &d);
```

(c)

```
char s[10];  
scanf("%c", s[2]);
```

->

```
char s[10];  
scanf("%c", &s[2]);
```

(d)

```
float f;  
scanf("%d", &f);
```

->

```
float f;  
scanf("%f", &f);
```

5.

(a) `afp = fopen("junk.txt", "a");`

(b) `pfp = fopen("para.txt", "w+");`

(c) `bfp = fopen("company.dat", "rb");`

6. (4)

7. (4)

8. `getchar()`

9. (2)

10.

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

```
struct test {
```

```
    int a;
```

```
    float b;
```

```
} record;
```

```
int main(void)
```

```
{
```

```
    FILE *fp=fopen("TEST.DAT", "rb");
```

```
    fread(&record, sizeof(record), 1, fp);
```

```
    fclose(fp);
```

```
    return 0;
```

```
}
```

11.

```
i = 8
```

프로그래밍 문제

1.

```

#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    FILE *fp1, *fp2;
    char file1[100], file2[100];

    printf("첫번째 파일 이름: ");
    scanf("%s", file1);

    printf("두번째 파일 이름: ");
    scanf("%s", file2);

    // 첫번째 파일을 읽기 모드로 연다.
    if( (fp1 = fopen(file1, "r")) == NULL )
    {
        fprintf(stderr, "원본 파일 %s을 열 수 없습니다.\n", file1);
        exit(1);
    }

    // 두번째 파일을 읽기 모드로 연다.
    if( (fp2 = fopen(file2, "r")) == NULL )
    {
        fprintf(stderr, "복사 파일 %s을 열 수 없습니다.\n", file2);
        exit(1);
    }

    // 첫 번째 파일과 두 번째 파일을 비교
    while( 1 ){
        int c1 = fgetc(fp1);
        int c2 = fgetc(fp2);
        if( c1 == EOF || c2 == EOF )
            break;
        if( c1 != c2 ) {
            printf("두 파일은 서로 다릅니다.\n");
            return 0;
        }
    }
    printf("두 파일은 서로 같습니다.\n");
}

```

```
        fclose(fp1);
        fclose(fp2);

    return 0;
}
```

2.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

```
int main(void)
{
    FILE *fp1, *fp2;
    char file1[100], file2[100];
    char c;

    printf("첫번째 파일 이름: ");
    scanf("%s", file1);

    printf("두번째 파일 이름: ");
    scanf("%s", file2);

    // 첫번째 파일을 읽기 모드로 연다.
    if( (fp1 = fopen(file1, "r")) == NULL )
    {
        fprintf(stderr, "파일 %s을 열 수 없습니다.\n", file1);
        exit(1);
    }

    // 두번째 파일을 읽기 모드로 연다.
    if( (fp2 = fopen(file2, "w")) == NULL )
    {
        fprintf(stderr, "파일 %s을 열 수 없습니다.\n", file2);
        exit(1);
    }

    // 첫 번째 파일과 두 번째 파일을 비교
    while( 1 ){
        c = fgetc(fp1);
```

```

        if( c== EOF)
            break;
        fputc(toupper(c), fp2);
    }
    fclose(fp1);
    fclose(fp2);
    return 0;
}

```

3.

// 이진 파일 버전

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

```
#define SIZE 100
```

```
int main(void)
```

```
{
```

```
    int buffer[SIZE];
```

```
    FILE *fp = NULL;
```

```
    FILE *fp1 = NULL;
```

```
    int i;
```

```
    int count;
```

```
    char file1[100], file2[100];
```

```
    printf("원본 파일 이름: ");
```

```
    scanf("%s", file1);
```

```
    printf("복사 파일 이름: ");
```

```
    scanf("%s", file2);
```

```
    fp = fopen(file1, "rb");          // ①
```

```
    if( fp == NULL )
```

```
{
```

```
        fprintf(stderr, "파일을 열 수 없습니다.");
```

```
        return 1;
```

```
}
```

```
    fp1 = fopen(file2, "wb");         // ①
```

```
    if( fp1 == NULL )
```

```
{
```

```
        fprintf(stderr, "파일을 열 수 없습니다.");
```

```
        return 1;
```

```
}
```



```

while( (count=fread(buffer, sizeof(char), SIZE, fp)) != 0 ){
    fwrite(buffer, sizeof(char), count, fp1);
}
fclose(fp);
fclose(fp1);
return 0;
}

```

4.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

```

```

int main(void)
{
    FILE *fp1, *fp2;
    char file1[100], file2[100];
    char buffer1[1000], buffer2[1000];

    printf("첫번째 파일 이름: ");
    scanf("%s", file1);

    printf("두번째 파일 이름: ");
    scanf("%s", file2);

    // 첫번째 파일을 읽기 모드로 연다.
    if( (fp1 = fopen(file1, "r")) == NULL )
    {
        fprintf(stderr, "원본 파일 %s을 열 수 없습니다.\n", file1);
        exit(1);
    }

    // 두번째 파일을 읽기 모드로 연다.
    if( (fp2 = fopen(file2, "r")) == NULL )
    {
        fprintf(stderr, "복사 파일 %s을 열 수 없습니다.\n", file2);
        exit(1);
    }

    // 첫번째 파일을 두번째 파일로 복사한다.
    while( 1 ){

```

```

        char *p1 = fgets(buffer1, 1000, fp1);
        char *p2 = fgets(buffer2, 1000, fp2);
        if( p1 == NULL || p2 == NULL )
            break;
        if( strcmp(buffer1, buffer2) != 0 ) {
            printf("<< %s", buffer1);
            printf(">> %s", buffer2);
        }
    }

    fclose(fp1);
    fclose(fp2);

    return 0;
}

```

5.

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

```
int main(void)
```

```

{
    FILE *fp = NULL;
    FILE *fpout = NULL;
    char buffer[1000]={0};
    char name[1000]={0};
    int korean, math, english;

    fp = fopen("sample.txt", "r");
    if( fp == NULL ){
        printf("파일 열기 실패\n");
        return 0;
    }
    fpout = fopen("sample1.txt", "w");
    if( fpout == NULL ){
        printf("파일 열기 실패\n");
        return 0;
    }
    while(fgets(buffer, 1000, fp) != NULL )
    {
        sscanf(buffer, "%s %d %d %d", name, &korean, &math, &english);
        fprintf(fpout, "%s", name);
    }
}

```

```

        fprintf(fpout, "%10.2f\n", (korean+math+english)/3.0);
    }
    fclose(fp);
    return 0;
}

```

6.

```

#include <stdio.h>
#include <ctype.h>

```

```

int main(void)
{
    FILE *fp = NULL;
    int c, count=0;
    char file1[100];

    printf("파일 이름을 입력하시오: ");
    scanf("%s", file1);

    // 첫번째 파일을 읽기 모드로 연다.
    if( (fp = fopen(file1, "r")) == NULL )
    {
        fprintf(stderr, "원본 파일 %s을 열 수 없습니다.\n", file1);
        exit(1);
    }
    while((c = fgetc(fp)) != EOF )
    {
        if( isprint(c) )
            count++;
    }
    fclose(fp);
    printf("문자의 개수는 %d\n", count);
    return 0;
}

```

7.

```

#include <stdio.h>
#include <stdlib.h>

```

```

int main(void)
{

```

```

FILE *fp1;
char file1[100];
char buffer[1000];

printf("파일 이름을 입력하시오: ");
scanf("%s", file1);

if( (fp1 = fopen(file1, "w")) == NULL )
{
    fprintf(stderr, "파일 %s을 열 수 없습니다.\n", file1);
    exit(1);
}

while( 1 ){
    char *p= gets(buffer);
    if( p == NULL )
        break;
    fprintf(fp1, "%s\n", buffer);
}

fclose(fp1);
return 0;
}

```

8.

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

```

int main(void)
{
    FILE *fp1 = NULL;
    FILE *fp2 = NULL;
    int value, i;

    fp1 = fopen("sample1.txt", "w");
    if( fp1 == NULL ){
        printf("파일 열기 실패\n");
        return 0;
    }
    fp2 = fopen("sample2.txt", "wb");
    if( fp2 == NULL ){
        printf("파일 열기 실패\n");
    }
}

```

```

        return 0;
    }
    for(i=0;i<100;i++)
    {
        value = rand();
        fprintf(fp1, "%d", value);
        fwrite(&value, sizeof(int), 1, fp2);
    }
    fclose(fp1);
    fclose(fp2);
    return 0;
}

```



sample1
sample2

txt
txt

462 2011-08-14 18:12 -a-
400 2011-08-14 18:12 -a-

9.

```

#include <stdio.h>
#include <stdlib.h>

```

```

int main(int argc, char *argv[])
{
    FILE *fp1, *fp2, *fp3;
    char file1[100], file2[100], file3[100];
    char buffer[100];

    if( argc >= 4 ){
        strcpy(file1, argv[1]);
        strcpy(file2, argv[2]);
        strcpy(file3, argv[3]);
    }
    else {
        printf("인수가 부족합니다.\n");
        return 0;
    }
    printf("%s 파일+ %s파일을 %s파일로 복사합니다.\n", file1, file2, file3);
    // 첫번째 파일을 읽기 모드로 연다.
    if( (fp1 = fopen(file1, "r")) == NULL )
    {
        fprintf(stderr, "원본 파일 %s을 열 수 없습니다.\n", file1);
        exit(1);
    }
}

```

```

// 두번째 파일을 읽기 모드로 연다.
if( (fp2 = fopen(file2, "r")) == NULL )
{
    fprintf(stderr, "원본 파일 %s을 열 수 없습니다.\n", file2);
    exit(1);
}
// 두번째 파일을 읽기 모드로 연다.
if( (fp3 = fopen(file3, "w")) == NULL )
{
    fprintf(stderr, "결과 파일 %s을 열 수 없습니다.\n", file3);
    exit(1);
}

// 첫번째 파일을 두번째 파일로 복사한다.
while( fgets(buffer, 100, fp1) != NULL )
    fputs(buffer, fp3);
// 첫번째 파일을 두번째 파일로 복사한다.
while( fgets(buffer, 100, fp2) != NULL )
    fputs(buffer, fp3);

fclose(fp1);
fclose(fp2);
fclose(fp3);

return 0;
}

```

10.

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

```
#include <ctype.h>
```

```
int main(void)
```

```

{
    FILE *fp = NULL;
    char buffer[500];
    int line_number=1;
    fp = fopen("sample.txt", "r");
    if( fp == NULL ){
        printf("파일 열기 실패\n");
        return 0;
    }
}

```

```

    }
    while(fgets(buffer, 500, fp)!=NULL)
    {
        printf("%d: %s\n", line_number, buffer);
        line_number++;
    }
    fclose(fp);
    return 0;
}

```

11.

```

#include <stdio.h>
#include <string.h>
#define TITLE_SIZE 50
#define NAME_SIZE 50
#define PUBLISHER_SIZE 50

typedef struct BOOK{
    char title[TITLE_SIZE];
    char author[NAME_SIZE];
    char publisher[PUBLISHER_SIZE];
} BOOK;

void add_record(BOOK library[], int count);
void menu();
int get_input();
void search_record(BOOK library[], int count);
void print_record(BOOK library[], int count);
int main(void)
{
    int num, count = 0;
    BOOK library[30] = {"\0"};
    while(1)
    {
        menu();
        num = get_input();
        switch(num)
        {
            case 1:
                add_record(library, count);
                count++;

```

```

        continue;
    case 2:
        print_record(library, count);
        continue;
    case 3:
        search_record(library, count);
        continue;
    case 4:
        return -1;
    }
    return 0;
}

void add_record(BOOK library[], int count)
{
    int type;

    fflush(stdin);
    printf("도서의 이름:");
    gets(library[count].title);
    printf("저자:");
    gets(library[count].author);
    printf("출판사:");
    gets(library[count].publisher);
}

void menu()
{
    printf("=====\n");
    printf(" 1. 추가\n");
    printf(" 2. 출력\n");
    printf(" 3. 검색\n");
    printf(" 4. 종료\n");
    printf("=====\n");
}

int get_input()
{
    int num;
    printf("정수값을 입력하시오 : ");
    scanf("%d",&num);
    return num;
}

```



```

void search_record(BOOK library[], int count)
{
    int i;
    char title[TITLE_SIZE];

    fflush(stdin);
    printf("제목: ");
    gets(title);

    for(i = 0; i < count; i++)
    {
        if(strcmp(title,library[i].title) == 0)
        {
            printf("출판사는 %s\n",library[i].publisher);
            return;
        }
    }
    printf("찾는 책이 테이블에 없습니다.\n");
}

void print_record(BOOK library[], int count)
{
    int i;
    fflush(stdin);
    for(i = 0; i < count; i++)
    {
        printf("제목 : %s\n",library[i].title);
        printf("저자 : %s\n",library[i].author);
        printf("출판사 : %s\n",library[i].publisher);
    }
}

```

12.

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

```

int main(void)
{
    FILE *fp = NULL;
    char fname[100];
    char word[100];
    char buffer[1000];
    int count=0;

```

```

printf("파일 이름: ");
scanf("%s", fname);
printf("탐색할 단어: ");
scanf("%s", word);

fp = fopen(fname, "r");
if( fp == NULL ){
    printf("파일 열기 실패\n");
    return 0;
}
while(fgets(buffer, 1000, fp) != NULL )
{
    count++;
    if( strstr(buffer, word) != NULL ){
        printf("%s:%d  %s\n", fname, count, buffer);
    }
}
fclose(fp);
return 0;
}

```

13.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(void)
{
    FILE *fp1, *fp2;
    char file1[100], file2[100];
    char buffer[100];

    printf("파일 이름: ");
    scanf("%s", file1);

    printf("파일 이름: ");
    scanf("%s", file2);

    // 첫번째 파일을 읽기 모드로 연다.
    if( (fp1 = fopen(file1, "r")) == NULL )

```

```

{
    fprintf(stderr, "파일 %s을 열 수 없습니다.\n", file1);
    exit(1);
}

// 두번째 파일을 쓰기 모드로 연다.
if( (fp2 = fopen(file2, "w")) == NULL )
{
    fprintf(stderr, "파일 %s을 열 수 없습니다.\n", file2);
    exit(1);
}

// 첫번째 파일을 두번째 파일로 복사한다.
while( fgets(buffer, 100, fp1) != NULL ) {
    char *pos = strtok(buffer, " ");
    while(pos != NULL) {
        if( strcmp("Android" , pos)==0 )
            fprintf(fp2, "안드로이드 ");
        else
            fprintf(fp2, "%s ", pos);
        pos = strtok(NULL, " ");
    }
}
fclose(fp1);
fclose(fp2);
return 0;
}

```

CHAPTER 17

중간점검 문제

p.699

1. 동적 메모리 할당
2. malloc()
3. free()
4. stdlib.h

p.704

1. calloc()
2. realloc()
3. 바이트
4. void *

p.715

1. 포인터
2. 데이터, 링크
3. 자기 참조 구조체
- 4.

배열(array): 크기가 고정되어 있다.

연결 리스트(linked list): 동적으로 크기를 늘릴 수 있다.

Summary

- 동적 메모리 할당

- malloc()
- 바이트
- void *
- free()
- stdlib.h

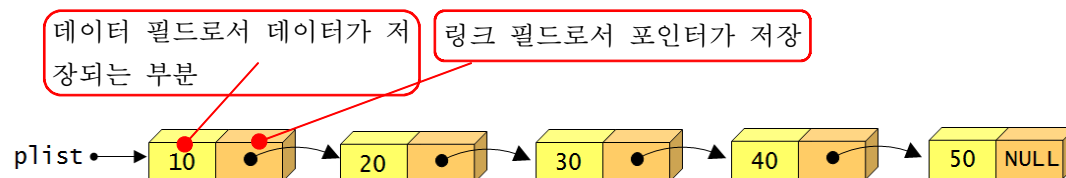
```
int *ptr;
ptr = (int *) malloc(10 * sizeof (int));
if (ptr == NULL) {
    ...
} else {
    ...
    free(ptr);
}
```

10개의 int를 동적으로 할당

반환된 포인터가 NULL이면 동적 메모리 할당 오류이다.

동적 메모리 할당을 해제한다.

- calloc()
- realloc()
- 할당받고자 하는 공간의 크기(바이트 단위)
- 변경되지 않는다.
- 물론 운영체제가 동적 메모리들을 회수하는 것이 일반적이지만 애플리케이션이 free()를 호출하는 것이 좋다.



- 데이터, 링크
 - 포인터
 - 배열(array): 크기가 고정되어 있다.
- 연결 리스트(linked list): 동적으로 크기를 늘릴 수 있다.

연습문제

1. (3)
- 2.

```
char* p;
p = (char *)malloc(1000);
```

```
printf("텍스트를 입력하시오: ");
gets(p);
printf("입력된 텍스트는 %s입니다. \n", p);
```

3.

- (a) malloc(), free()
- (b) void *
- (c) size_t

4.

(a)
int *pi;
pi = (int *)malloc(sizeof(int)*10);

(b)
char *pi;
pi = (char *)calloc(10, sizeof(char));

(c)
char *pc;
pc = (char *)malloc(10*sizeof(char));
또는
double *pc;
pc = (double *)malloc(10*sizeof(double));

5.

(a)
double *pd;
pd = (double *)malloc(sizeof(double));
*pd = 3.14;

(b)

int *pi;
pi = (int *)malloc(sizeof(int)*5);
pi[0] = 1;
pi[1] = 2;
pi[2] = 3;
pi[3] = 4;
pi[4] = 5;

6.

- (a) 포인터 p가 다음 노드를 가리키도록 함
- (b) 현재 노드의 다음, 다음 노드
- (c) plist가 NULL이 아니라면 반복이 끝난 후에 p는 모두 연결 리스트의 마지막 노드를 가

리킨다.

7. (2)

프로그래밍 문제

1.

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

```
int main(void)
```

```
{
```

```
    int count, i;
    int value, sum;
    int *darray;
```

```
    printf("정수의 개수: ");
    scanf("%d", &count);
```

```
    darray = (int *)malloc(count*sizeof(int));
    if( darray == NULL ){
        printf("동적 메모리 할당 오류");
        exit(1);
    }
```

```
    for(i=0;i<count;i++){
        printf("양의 정수를 입력하시오: ");
        scanf("%d", &darray[i]);
    }
```

```
    sum = 0;
    for(i=0;i<count;i++){
        sum += darray[i];
```

```

    }
    printf("합은 %d입니다.\n", sum);

    return 0;
}

```

2. 실행 결과 오타!

문자열의 개수: 10
 10개의 문자열을 저장할 수 있는 공간이 할당되었습니다.

HINT 오타!

HINT 문자열은 `char*`로 가리킬 수 있다. 문자열의 배열은 `char**`로 가리킬 수 있다. 다음의 코드를 참조하라.

```

char** A = NULL;
if ((A = malloc(n*sizeof(char*))) != NULL) {           // 먼저 n개의 포인터를 위한 공간을
    할당한다.
        for (i=0;i<n;i++)
            A[i] = (char*)malloc(100); // 100개의 문자를 저장할 수 있는 공간 할당
    }

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

int main(void)
{
    int n, i;
    char **A;

    printf("문자열의 개수: ");
    scanf("%d", &n);

    if ((A = (char **)malloc(n*sizeof(char*))) != NULL) {
        for (i=0;i<n;i++)
            A[i] = (char*)malloc(100);
    }
    for (i=0;i<n;i++)
        sprintf(A[i], "%d", i);

    for (i=0;i<n;i++)
        printf("%s \n", A[i]);
}

```



```

        for (i=0;i<n;i++)
            free(A[i]);
        free(A);

        return 0;
    }

```

3.

```

#include <stdio.h>
#include <stdlib.h>

```

```

typedef struct contacts { // 구조체 타입 정의
    char name[100];
    char mobile_phone[100];
} CONTACTS;

```

```

int main(void)
{

```

```

    CONTACTS *contacts;    // 동적 메모리 공간을 가리키는 포인터
    int size, i;

```

```

    printf("주소의 개수: ");
    scanf("%d", &size);

```

```

    contacts = (CONTACTS *)malloc(sizeof(CONTACTS)* size);    // 동적 메모리 할당
    if( contacts == NULL ){
        printf("동적 메모리 할당 오류");
        exit(1);
    }

```

```

    for(i=0; i<size ;i++) {
        printf("이름을 입력하시오: ");
        fflush(stdin);           // 입력 버퍼를 비운다.
        gets(contacts[i].name);
        printf("휴대폰 번호를 입력하시오: ");
        gets(contacts[i].mobile_phone);
    }
    printf("=====\n");

```

```

        printf("이름          휴대폰 번호\n");
        printf("=====\n");
        for(i=0;i<size;i++)
            printf("%s %s \n", contacts[i].name, contacts[i].mobile_phone);
        printf("\n=====\n");
        free(contacts);          // 동적 메모리 공간 해제
        return 0;
    }

```

4.

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

```
#include <stdlib.h>
```

```

char* get_word() {
    char* s = (char *)malloc(50);
    if (s == NULL) {
        printf("메모리 할당 오류\n");
        exit( 1 );
    }
    printf("단어를 입력하시오(최대 50글자:");
    scanf("%s", s);
    return s;
}

```

```
int main(void)
```

```

{
    char *p;
    p = get_word();
    free(p);
    return 0;
}

```

5.

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

```
#include <malloc.h>
```

```
typedef int DATA;
```

```
typedef struct NODE
```

```

{
    DATA data;
    struct NODE *link;
}NODE;

```

```

void print_list(NODE *plist);
int get_integer();
int main(void)
{
    NODE *plist;
    NODE *curr = NULL, *prev = NULL;
    int i;
    plist = NULL;
    while(1)
    {
        i = get_integer();
        if(i < 0)
            break;
        curr = (NODE *)malloc(sizeof(NODE));
        curr->data = i;
        curr->link = NULL;
        if(prev == NULL)
            plist = curr;
        else prev->link = curr;
        prev = curr;
    }
    print_list(plist);
    return 0;
}

void print_list(NODE *plist)
{
    NODE *p;
    p = plist;
    while(p)
    {
        printf("%d->",p->data);
        p = p->link;
    }
    printf("NULL\n");
}

int get_integer()
{
    int num;
    printf("양의 정수를 입력하세요(종료 -1): ");
    scanf("%d",&num);
}

```

```
        return num;
    }
```

6.

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

```
#define TRUE    1
#define FALSE   0
```

```
typedef struct
{
    char name[30];
    char phone[20];
}DATA;
```

```
typedef struct NODE
{
    DATA data;
    struct NODE *link;
}NODE;
```

```
NODE *search_pos(NODE *plist, char name[]);
NODE *search(NODE *plist, char name[]);
NODE *create_node(char *name, char* phone, NODE *link);
void insert_node(NODE **phead,  NODE *node, NODE *prev);
void print_menu();
```

```
int main(void)
{
    char name_t[30]={'\0'};
    char phone_t[20]={'\0'};
    char cmenu=1;
    int bEscape = FALSE;
    NODE* List = NULL;
    NODE* Temp = NULL;

    while(TRUE)
    {
```

```

fflush(stdin);
print_menu();
printf("번호를 입력하세요 : ");
scanf("%c", &cmenu);
switch(cmenu)
{
case '1':
    break;
case '2':
    printf("이름 : ");
    scanf("%s", name_t);
    printf("번호 : ");
    scanf("%s", phone_t);
    insert_node(&List, create_node(name_t, phone_t, NULL),
search_pos(List, name_t));
    printf("추가되었습니다.\n");
    break;
case '3':
    printf("찾을 이름을 입력하세요 : ");
    scanf("%s", name_t);
    Temp = search(List, name_t);
    if(Temp!=NULL)
    {
        printf("전화 번호부를 찾았습니다.\n");
        printf("이름 : %s\n번호 : %s\n", Temp->data.name,
Temp->data.phone);
    }
    else
        printf("데이터를 찾지 못했습니다.\n");
    break;
case '4':
    printf("찾을 이름을 입력하세요 : ");
    scanf("%s", name_t);
    Temp = search(List, name_t);
    printf("변경할 이름을 입력하세요 : ");
    scanf("%s", name_t);
    printf("변경할 번호를 입력하세요 : ");
    scanf("%s", phone_t);
    strcpy(Temp->data.name, name_t);
    strcpy(Temp->data.phone, phone_t);
    printf("데이터가 변경되었습니다.\n");

```

```

        break;
    case '5':
        bEscape = TRUE;
        break;
    default:
        printf("잘못 입력 하셨습니다. 다시입력해주세요.\n");
        break;
}

if(bEscape)
    break;
}

return 0;
}

```

```

void print_menu()
{
    printf("-----\n");
    printf("1. 초기화\n");
    printf("2. 전화 번호 추가\n");
    printf("3. 전화 번호 탐색\n");
    printf("4. 전화 번호 변경\n");
    printf("5. 종료\n");
    printf("-----\n");
}

```

```

NODE *search_pos(NODE *plist, char name[])
{
    NODE *curr, *prev;
    prev = NULL;
    curr = plist;
    while(curr != NULL)
    {
        if(strcmp(curr->data.name, name) > 0)
            break;
        prev = curr;
        curr = curr->link;
    }
    return prev;
}

```

```
}
```

```
NODE *search(NODE *plist, char name[])
```

```
{
```

```
    NODE *curr;
```

```
    curr = plist;
```

```
    while(curr != NULL)
```

```
    {
```

```
        if(strcmp(curr->data.name, name) == 0)
```

```
            break;
```

```
        curr = curr->link;
```

```
    }
```

```
    return curr;
```

```
}
```

```
NODE *create_node(char *name, char* phone, NODE *link)
```

```
{
```

```
    NODE *new_node;
```

```
    new_node = (NODE *)malloc(sizeof(NODE));
```

```
    if( new_node == NULL ) {
```

```
        fprintf(stderr, "메모리 할당 에러\n");
```

```
        exit(1);
```

```
    }
```

```
    strcpy(new_node->data.name, name);
```

```
    strcpy(new_node->data.phone, phone);
```

```
    new_node->link = link;
```

```
    return(new_node);
```

```
}
```

```
void insert_node(NODE **phead, NODE *node, NODE *prev)
```

```
{
```

```
    if( *phead == NULL )
```

```
    {
```

```
        *phead = node;
```

```
        node->link = NULL;
```

```
    }
```

```
    else if(prev == NULL)
```

```
    {
```

```
        node->link = (*phead)->link;
```

```
        (*phead) = node;
```

```
    }
```

```

        else
        {
            node->link = prev->link;
            prev->link = node;
        }
    }
}

```

7.

```

#include <stdio.h>
#include <stdlib.h>

```

```

#define INSERT  1
#define PRINT   2
#define QUIT    3

```

```

#define S_SIZE 50

```

```

typedef struct data{
    char title[S_SIZE];
    int year;
}DATA;
typedef struct NODE{
    DATA data;
    struct NODE *link;
}NODE;

```

```

NODE *insert_node(NODE *plist,NODE *pprev,DATA item)
{
    NODE *pnew = NULL;
    if(!(pnew = (NODE*)malloc(sizeof(NODE))))
    {
        printf("메모리 동적 할당 오류\n");
        exit(1);
    }

    pnew->data = item;
    if(pprev == NULL)
    {
        pnew->link = plist;
        plist = pnew;
    }
}

```



```

        else
        {
            pnew->link = pprev->link;
            pprev->link = pnew;
        }
        return plist;
    }

void display_menu()
{
    printf("-----\n");
    printf("1. 영화 정보 추가\n");
    printf("2. 영화 정보 출력\n");
    printf("3. 종료\n");
    printf("-----\n");
}

int get_selectde_menu()
{
    int selmenu = 0;
    printf("번호를 선택하시오 : ");
    scanf("%d", &selmenu);
    return selmenu;
}

DATA get_input()
{
    DATA input;

    printf("영화의 제목을 입력하시오 : ");
    scanf("%s", input.title);
    printf("영화의 개봉 연도를 입력하시오 : ");
    scanf("%d", &input.year);

    return input;
}

void print_list(NODE *plist)
{
    NODE *ptr;

```

```

    for(ptr=plist;ptr;ptr=ptr->link)
    {
        printf("-----\n");
        printf("제      목 : %s\n", ptr->data.title);
        printf("개봉 연도 : %d\n", ptr->data.year);
    }
    printf("-----\n");
}

```

```

void destroy_nodes(NODE *plist)
{
    NODE *temp;
    while(plist)
    {
        temp = plist;
        plist = plist->link;
        free(temp);
    }
}

```

```

int main(void)
{
    NODE *plist = NULL;
    int selected = 0;
    DATA d;

    while(selected != QUIT)
    {
        display_menu();
        selected = get_selectde_menu();

        switch(selected)
        {
            case INSERT:
                d = get_input();
                plist = insert_node(plist,NULL,d);
                break;
            case PRINT:
                print_list(plist);
                break;
            case QUIT:

```

```
        destroy_nodes(plist);  
        break;  
    }  
}  
  
return 0;  
}
```

감사합니다.

END