

개정3판

Visual
Studio
2017

쉽게 풀어쓴 C언어 EXPRESS

천인국 지음



홈페이지 www.bodan.co.kr 제공

프로그래밍 초소

면접준비로 프로그래밍 문제 풀기 용이

쉽고 재미있는 프로그래밍의 세계로!

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

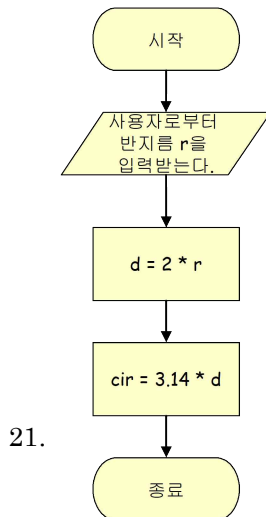


CHAPTER 1

연습문제

1. (1) 2진법
3. (2) 순서도
5. (1) 알고리즘
7. (2)
9. 이미지의 경우, 각 픽셀의 밝기값이 숫자로 표현된다. 음악의 경우, 음파의 높이가 숫자로 표현된다. 좀 더 자세한 내용을 인터넷에서 조사하여 본다.
11.
 - * 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
13.
 - * 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
15.
 - * 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
17. 미리 정해진 특정 기능을 수행하기 위해 컴퓨터의 하드웨어와 소프트웨어가 조합된 전자 제어 시스템을 말한다.

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



23. 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.

25.

- (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

연습문제

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

3.
 - (a) `printf(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");`

프로그래밍

1.

```
#include <stdio.h>

int main(void)
{
    printf("이름: 홍길동\n");
    printf("나이: 21살\n");
    printf("주소: 서울 200번지\n");
    return 0;
}
```

3.

```
#include<stdio.h>

int main(void)
{
    printf("*****\n");
    printf("                리포트\n");
    printf("*****\n\n");
    printf("*****\n");
    printf("                학과: 컴퓨터공학과\n");
    printf("                학번: 20180001\n");
    printf("                성명: 홍길동\n");
    printf("*****\n\n");
    return 0;
}
```

5.

```
#include <stdio.h>

int main(void)
{
    printf("7+8=%d\n", 7 + 8);
    printf("7-8=%d\n", 7 - 8);
    printf("7*8=%d\n", 7 * 8);
    printf("7/8=%d\n", 7 / 8);
    return 0;
}
```

CHAPTER 3

연습문제

1. (3)
3. (3)
5. (4)
- 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;
}
```

- 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 . . .
```

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 . . .

```

5.

```

#include<stdio.h>
int main(void)
{
    double num, ans;
    printf("실수를 입력하세요: ");
    scanf("%lf" , &num);
    ans = 3.0*num*num+7.0*num+11;
    printf("다항식의 값은 %f\n", ans);
    return 0;
}

```

```

실수를 입력하시요: 2.0
다항식의 값은 37.000000
계속하려면 아무 키나 누르십시오 . . .

```


CHAPTER 4

연습 문제

1. (3) 숫자로 시작하면 안 된다.

3. (3)

5. (3)

7. (4)

9.

(a) 잘못없음

(b) #는 사용할 수 없는 기호

(c) 숫자로 시작할 수 없음

(d) %를 사용할 수 없음

11. (1), (3), (5)

13. (a) %f 또는 %lf (b) %d (c) %c (d) %f

15.

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

```
int main(void)
```

```
{
```

```
    int x, y = 0;
```

```
    char grade = "A";
```

```
    double rate = e10;
```

```
    short double profit = 75.0;
```

```
    int salary = 2,000,000;
```

```
}
```

```
// int x=0, y=0;
```

```
// char grade = 'A';
```

```
// double rate = 1.0e10;
```

```
// double profit = 75.0;
```

```
// 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;
}
```

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;
}
```

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. \n",m);

    return 0;
}
```

7.

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

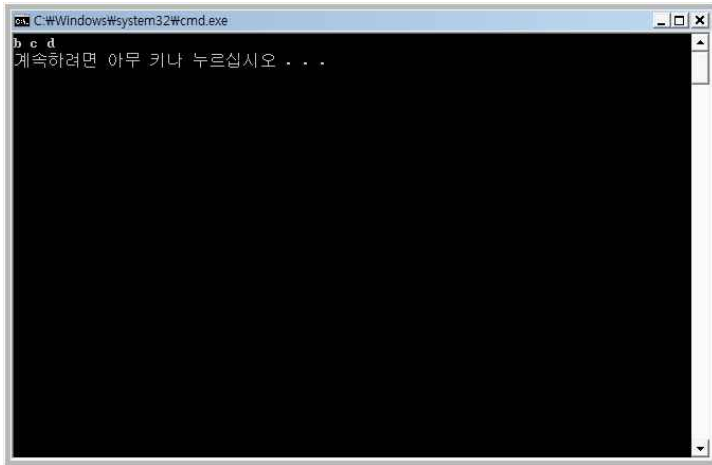
    printf("질량(kg): ");
    scanf("%lf", &mass);
    printf("속도(m/s): ");
    scanf("%lf", &speed);

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

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

9.

```
#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.

11.

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

CHAPTER 5

연습문제

1. (2)
3. (1)
5. (a) 1.5 (b) 1.0 (c) 1.5 (d) 1.5
7. -2, 3, 1, 1
9. (a) 0 (b) 0xffff (c) 0xffff (d) x의 모든 비트가 반전된다.
11. 80
13.
 - (a) `(years >= 3) && (age >= 40) && (family_members >= 3)`
 - (b) `(age >= 6) && ((height >= 150) || ((height < 150) && (has_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

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 \n", max);

}
```

3개의 정수를 입력하시오:2 3 4
최대값:4

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;
}
```

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;
}
```

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;
    }
```

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 \n", radius);

    return 0;
}
```

```
거리를 입력하시오:900
각도를 입력하시오:7.2
지구의 반지름은 7165.605096
```


CHAPTER 6

연습문제

1. (1)
3. (2), (3)
- 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;
```

7.

(a)

0

1

(b)

C



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;
}
```

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;
}

```

5.

```

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

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

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

```

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;
}

```

11.

```

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

```

```
    return 0;  
}
```

CHAPTER 7

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.

3. 한번도 출력되지 않는다.

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);
}
```

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", x);

```

(e) 변수 i가 초기화되지 않았다.

```

int i;
for( i=0 ; i < 10; i++)
    printf("i = %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.

```
#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("\n\na");

    return 0;
}
```

3.

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

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

    printf("약수: ");
    for(int i=1;i<=x;i++){
        if( x%i==0)
            printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

5.

```
#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;
}
```

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:      ;
    }
    printf("\n");
    return 0;
}
```

9.

```
#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.\n", (i-1), sum-i);
return 0;
}

```

1부터 140까지의 합이 9870입니다.

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.\n", result);
    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("순열의 값은 %d입니다.\n", result);
    return 0;
}
```

CHAPTER 8

연습문제

1. (1)
3. (1), (2), (4)
5. (a) 1.720000 (b) 1.000000 (c) 2.000000
7.
 - (a) $y = \log_{10}(x) + \exp(x)$;
 - (b) $y = \sin(x) + \sqrt{x^2 - 2.0 \cdot a} + \text{pow}(2.0, 10)$;
9.

```
int f(void) ----- return 10 + 20;
void g(int, int) ----- return;
double h(double, int); ----- return 'a' + 1.0;
```
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);`
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 정의의 매개 변수 타입이 서로 다르다.
 - (d) `sum = x + y -> return x + y;`
 - (e) 반환형이 정의되지 않았는데 값을 반환하였다.

프로그래밍 문제

1.

```
#include<stdio.h>

double square(double x);

int main()
{
    double m, n;
    printf("정수를 입력하시오: ");
    scanf("%lf", &m);
    n = square(m);
    printf("주어진 정수 %f의 제곱은 %f입니다. \n", m, n);
}

double square(double x)
{
    double p;
    p = x * x;
    return (p);
}
```

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;
}

int main(void)
```

```

{
    double r;
    r = get_radius();
    printf("원의 면적은 %f입니다.\n", cal_area(r));

    return 0;
}

```

5.

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

```

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

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

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

    return 0;
}

```

```

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

```

7.

```
#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) (1000*0.08+(income-1000)*0.1);
    else return (int) (income*0.08);
}

```

9.

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

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

```
int b_rand();
```

```
int main(void)
```

```

{
    for(int i = 0; i<10; i++)
        printf("%d ", b_rand());
    printf("\n");
    return 0;
}

```

```
int b_rand()
```

```

{
    return rand() % 2;
}

```

11.

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

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

```
double f_rand();
```

```
int main(void)
```

```

{
    for(int i=0;i<5;i++)
        printf("%f ", f_rand());
    printf("\n");
    return 0;
}

```

```
double f_rand()
```

```
{
```



```

        return rand() / (double)RAND_MAX;
    }

```

13.

```

#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;
}

```

15.

```

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

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

    return 0;
}

```

```

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

```

17.

```

#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("두 개의 실수는 서로 같음\n");
    else
        printf("두 개의 실수는 서로 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;  
}
```

CHAPTER 9

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)
    ...
}
```

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("%i",
            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

5.

(a)

5
4
3
2
1
0

반환값은 16

(b)

5
4
3
2
1
0

반환값은 95

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);
    }
}

```

3.

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

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

```
int check()
```

```

{
    static int call_count = 0;
    call_count++;
    if( call_count >3 ){
        printf("로그인 시도횟수 초과\n ");
        return 0;
    }
    printf("비밀번호: ");
    int n;
    scanf("%d", &n);
    if( n == 1234 ){
        printf("로그인 성공!!\n ");
    }
}

```



```

        return 1;
    }
    return 0;
}
int main(void)
{
    check();
    check();
    check();
    check();
    return 0;
}

```

5.

```

#include <stdio.h>
int sum(int n);

int main()
{
    int number, result;

    printf("정수를 입력하시오: ");
    scanf("%d", &number);

    result = sum(number);

    printf("1부터 %d까지의 합=%d\n", number, result);
}

int sum(int num)
{
    if (num != 0)
        return num + sum(num - 1);
    else
        return num;
}

```

7.

```

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

```

9.

```

#include <stdio.h>
int get_digit_sum(int num);
int main()
{
    int n, sum;
    printf("정수를 입력하시오: ");
    scanf("%d", &n);
    sum = get_digit_sum(n);

    printf("자리수의 합: %d\n", sum);

    return 0;
}

```

```

int get_digit_sum(int n)
{
    if (n == 0)
        return 0;

    return ((n % 10) + get_digit_sum(n / 10));
}

```

11.

```

#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("n=");
    int n;
    scanf("%d", &n);
    printf("k=");
    int k;
    scanf("%d", &k);
    printf("%d\n", recursive(n, k));
    return 0;
}
```

CHAPTER 10

연습문제

1. (2)

3. (3), (4)

5. (4)

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]++;
    }
}

```

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;
}
```

3.

```
#include <stdio.h>

#define N_DATA 10

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

int main(void)
{
    int A[N_DATA] = { 1, 2, 3 };
    int B[N_DATA] = { 0 };
    array_print(A, 10);
    array_print(B, 10);
    if (array_equal(A, B, N_DATA) == 1)
        printf("2개의 배열은 같음 \n");
    else
        printf("2개의 배열은 다름 \n");
    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;
}

```

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("가장 많이 나온수=%d\n", max);
    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;
}

```

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("시험 #%d의 최대점수=%d\n", k, max);
        printf("시험 #%d의 최저점수=%d\n", k, 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;
}

```


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;
}
```

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;
}
```

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;
}
```

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

int main()
{
    int board[10][10] = { 0 };
    srand(time(NULL));

    for (int i = 0; i < 10; i++)
        for (int j = 0; j < 10; j++)
            if ((rand() % 100) < 30)
                board[i][j] = 1;
```

```
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 10; j++)
                if (board[i][j])
                    printf("# ");
                else
                    printf(". ");
            printf("\n");
        }
    return 0;
}
```

CHAPTER 11

연습문제

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

3. (3)

5.

5

5

계속하려면 아무 키나 누르십시오 . . .

7.

0

1

1

계속하려면 아무 키나 누르십시오 . . .

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.

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");
}

```

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("%05.2f ", a[i]);
    }
    printf("\n");
}
int main(void)
{
    double grades[10] = { 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.3 };
    double scores[10] = { 0 };
    print(grades, 10);
    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);
    }
}

```

7.

```

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

```

```

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

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

    array_print("A", A, N_DATA);
    array_print("B", B, N_DATA);
    array_add(A, B, C, N_DATA);
    array_print("C", 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];
    }
}

```

9.

```

#include <stdio.h>

int search(int *A, int size, int x);

int main(void)
{
    int data[10] = { 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 };
    int index = search(data, 10, 200);
    printf("월급이 200만원인 사람의 인덱스=%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;
}

```

11.

```

#include <stdio.h>
#define N_DATA 4
void merge(int *A, int *B, int *C, int size);
void array_print(char *name, int *a, int size)
{
    int i;
    printf("%s[] = ", name);
    for (i = 0; i<size; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");
}
int main(void)
{
    int i;

    int A[] = { 2, 5, 7, 8};
    int B[] = { 1, 3, 4, 6};
    int C[8];
    array_print("A", A, N_DATA);
    array_print("B", B, N_DATA);
    merge(A, B, C, 4);
    array_print("C", C, 2*N_DATA);
    for(i=0;i<8;i++)
        printf("%d ", C[i]);
    return 0;
}

void merge(int *A, int *B, int *C, int size)
{
    int i, a, b, c;

```



```

for(a=0, b=0, c=0; a<size && b<size;){
    if(A[a]<=B[b])
        C[c++] = A[a++];
    else
        C[c++] = B[b++];
}
for(i=a; i<size; i++)
    C[c++] = A[i];
for(i=b; i<size; i++)
    C[c++] = B[i];
}

```

1 2 3 4 5 6 7 8 계속하려면 아무 키나 누르십시오 . . .

CHAPTER 12

연습문제

1.

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

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);
```

5. (4)

7.

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

9.

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

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;
}
```

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 \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))
            printf("%c \n", toupper(c));
        if (isupper(c))
            printf("%c \n", tolower(c));
        if (!isalpha(c))
            printf("경고 ");
        if (c == '.') break;
        getchar();// 줄바꿈 문자 제거
    }
    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 \n");
    else
        printf("부정적인 답변 \n");
    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("수정된 텍스트:%s\n", s);
    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;

    printf("문자열을 입력하시오: ");
    gets_s(s, 100);

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

    while (ptr[i] != NULL)
    {
        ptr[i + 1] = strtok(NULL, " ");
        i++;
    }

    for (j = i - 1; j >= 0; j--)
    {
        printf("%s ", ptr[j]);
    }
    printf("\n ");
    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_s(str, SIZE);

    printf("구두점의 개수는 %d입니다.\n", get_punc(str));
    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 입력하시오:");
    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 \n", x + y);
    }
    else if (strcmp(op, "sub") == 0) {

```

```
        printf("연산의 결과: %d \n", x - y);
    }
    else if (strcmp(op, "mul") == 0) {
        printf("연산의 결과: %d \n", x*y);
    }
    else if (strcmp(op, "div") == 0) {
        printf("연산의 결과: %d \n", x / y);
    }
    else {
    }

    return 0;
}
```


CHAPTER 13

연습문제

1.

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

3. (2)

5.

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

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

7. 공용체에서는 모든 멤버를 동시에 초기화할 수 없다.

9.

(a)

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

(b)

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

프로그래밍 문제

1.

```
#include <stdio.h>

struct book {
    int id;
    char title[100];
    char author[20];
};

int main(void)
{
    struct book b1 = {1, "바람과 함께 사라지다", "마가렛 미첼"};
    printf("{ %d, %s, %s } \n", b1.id, b1.title, b1.author);
    return 0;
}
```

3.

```
#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;
};

print(struct email e)
{
    printf("제목: %s\n", e.title);
    printf("수신자: %s\n", e.receiver);
    printf("발신자: %s\n", e.sender);
    printf("내용: %s\n", e.content);
    printf("날짜: %s\n", e.date);
    printf("우선순위: %d\n", e.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;
```

```
    print(e);
```

```
    return 0;
```

```
}
```

5.

```
#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;
```

```
}
```

7.

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

```
struct point {
```

```
    int x, y;
```

```
};
```

```

int quadrant(struct point *p);
int main(void)
{
    struct point p = { -1, 2 };
    printf("(%d, %d)의 사분면 = %d\n", p.x, p.y, 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;
}

```

9.

```

#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("총 칼로리=%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;
}

```

11.

```
#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("집전화번호: %s\n", list[i].home_phone);
            printf("휴대폰번호: %s\n", list[i].cell_phone);
            return 0;
        }
    }
    printf("검색이 실패하였슴\n");

    return 0;
}
```

13.

```
#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;
}

```

CHAPTER 14

연습문제

1. 다음은 무엇을 선언하는 문장인가?

(a) `int **dp;`

int형 이중 포인터 변수 dp 선언

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

int형 포인터 배열 pa 선언

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

int형 배열 포인터 p 선언

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

int형 매개 변수를 가지고 반환형이 없는 함수를 가리키는 포인터 f 선언

3. 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
```

(d)

```
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("몇 번째 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];
```

```
}
```

3.

```
#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;
}
```

5.

```
#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];
        }
    }
}

7.
#include <stdio.h>
#define WIDTH 10
#define HEIGHT 10
void print_image(int p[HEIGHT][WIDTH], int h, int w)
{
    int x, y;
    for (y = 0; y < h; y++){
        for (x = 0; x < w; x++){
            printf("%d ", p[y][x]);
        }
        printf("\n");
    }
}

void reverse_image(int p[HEIGHT][WIDTH], int h, int w);
int main(void)
{
    int image[HEIGHT][WIDTH] = {
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,255,255,0,0,0,0,0},
        {0,0,255,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,0,0,255,0,0,0,0,0},
        {0,0,255,255,255,255,255,0,0,0}};
    reverse_image(image, HEIGHT, WIDTH);
    print_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];
    }

```

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

연습문제

1. (4)

3. (3)

5.

(a) `afp = fopen("junk.txt", "a");`

(b) `pfp = fopen("para.txt", "w+");`

(c) `bfp = fopen("company.dat", "rb");`

7. (4)

9. (2)

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;
}

```

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;
}

```

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;
    }
}
```

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;
```

```
}
```

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;
    }

```

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;

    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);
    getchar();
    return num;
}

void search_record(BOOK library[], int count)
{
    int i;
    char title[TITLE_SIZE];

```

```

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);
    }
}

```

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 16

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
```

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!

5. #define AREA(w, h) w*h -> #define AREA(w, h) ((w)*(h))

7.

(a)

firstsecond

(b)

Kim and Park Hello!

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

```

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);

    printf("[ ");
    for (int i = 0; i < 10; i++)
        printf("%d ", a[i]);
    printf("] \n");
    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;
}

```

7.

```
#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;
}
```

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;
}
```

CHAPTER 17

연습문제

1. (3)

3.

(a) malloc(), free()

(b) void *

(c) size_t

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;
```

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;
}
```


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);
```

```
    getchar();
```

```
    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: ");
```

```
        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;
```

```
}
```

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;
}

```

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);
    getchar();
    return selmenu;
}

DATA get_input()
{
    DATA input;

    printf("영화의 제목을 입력하시오 : ");
    gets_s(input.title, 50);
    printf("영화의 개봉 연도를 입력하시오 : ");
    scanf("%d", &input.year);
}

```

```

        getchar();
        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;

    printf("연결 리스트를 이용한 영화관리 메뉴\n");
    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