

2019142037 이지원

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<conio.h>
#include <ctype.h>
#include <string.h>
#define _CRT_SECURE_NO_WARNINGS
#define DT 3
#define PT 7
#define CP 10
#define DT 5
#define AL 5
#define AL 7
#define AL 8
#define R 3
#define C 5
#define N 20
#define R 2
#define C 3
#define R 3
#define C 3
#define Y 3
#define P 5
#define R 2
#define C 3
#define P 2
#define R 3
#define C 5
#define P 2
#define R 2
#define C 3
#define K 2
#define M 7
#define CR 2
#define CC 9
#define R 5
#define ST 5
#define ST 2
int sum(int sale[Y][P]); // 10 -16
```

```

void grid(); // 10 -15
int b_s(int list[], int n, int input); // 10-12
int sv(int score[], int size); //10-8
int gob(int ta); //10-9
int hap(int a, int b); //12-05
int cha(int a, int b); //12-05
void menu(void); //12-06
int hap(int a, int b); //12-06
int cha(int a, int b); //12 -06
int gob(int a, int b); //12-06
int na(int a, int b); //12 -6
void creed(char** ppc); // 12-08
void target(char** ppc); //12 -08
void bubble(char*, int); //12- 10
void selection(char*, int); //12 -11
int sequential(char*, int, char); //12-12
int binary(char*, int, char); //12-13

```

```

int binary(char* item, int count, char key)
{
    int low = 0, high, mid;
    high = count - 1;

    printf("\n[이진 탐색 함수 영역]\n\n");

    while (low <= high)
    {
        mid = (low + high) / 2;

        if (key < item[mid])
            high = mid - 1;
        else
            if (key > item[mid])
                low = mid + 1;
            else
                return mid;
    }
    return -1;
}

```

//위 사용자 정의 함수는 12-13

```
int sequential(char* item, int count, char key)
{
    int i;
    printf("\n[순차 탐색 함수 영역]\n\n");
    for (i = 0; i < count; ++i)

        if (key == item[i])
            return i;

        else

            return -1;
}
```

//위 사용자 정의 함수는 12-12

```
void selection(char* item, int count)
{
    int i, j, k;
    char t;

    printf("\n[선택 정렬 함수 영역]\n\n");
    for (i = 0; i < count - 1; ++i)
    {
        t = item[i];
        k = i;
        for (j = i + 1; j < count; ++j)
        {
            if (item[j] < t)
            {
                k = j;
                t = item[j];
            }
        }
        item[k] = item[i];
        item[i] = t;
    }
}
```

//위 사용자 정의 함수는 12-11

```

void bubble(char* item, int count)
{
    int pass = 0;
    int change = 1;
    int j;
    char t;
    printf("\n[버블 정렬 함수 영역] \n\n");

    while (pass < count - 1 && change)
    {
        change = 0;
        for (j = 0; j < count - pass - 1; ++j)
        {
            if (item[j] > item[j + 1]) // 오름차순 정렬 큰값의 문자로 교체
            {
                change = 1;
                t = item[j];
                item[j] = item[j + 1];
                item[j + 1] = t;
            }
        }

        pass++;
    }
}

```

//위 사용자 정의 함수는 12-10

```

void creed(char** ppc) {
    *ppc = "오늘 하루도 최선을 다하자.";
}

```

// 위 사용자 정의 함수는 12-08

```

void target(char** ppc) {
    *ppc = "후회하지 않는 삶을 살자.";
}

```

// 위 사용자 정의 함수는 12-08

```

void menu(void) {
    int size, cnt;
    const char* ma[] = { "덧셈", "뺄셈", "곱셈", "나눗셈", "종료" };
    size = sizeof(ma) / sizeof(ma[0]); // 배열 요소의 개수 산출
}

```

```

        for (cnt = 0; cnt < size; cnt++)
        {
            printf("%d.%s \n", cnt, ma[cnt]);

        }

    }

//위 사용자 정의 함수는 12-06
int hap(int a, int b)
{
    int result;
    printf("\n[덧셈 hap() 함수 영역]\n\n");
    result = a + b;
    return result;
}

//위 사용자 정의 함수는 12-06
int cha(int a, int b)
{
    int result;
    printf("\n[뺄셈 cha() 함수 영역]\n\n");
    result = a - b;
    return result;
}

//위 사용자 정의 함수는 12-06
int gob(int a, int b)
{
    int result;
    printf("\n[곱셈 gob() 함수 영역]\n\n");
    result = a * b;
    return result;
}

//위 사용자 정의 함수는 12-06
int na(int a, int b)
{
    int result;
    printf("\n[나눗셈 na() 함수 영역]\n\n");
    result = a / b;
    return result;
}

//위 사용자 정의 함수는 12-06
int hap(int a, int b) {

```

```

        int sum;
        printf("\n [덧셈 hap() 함수 영역] \n\n");
        sum = a + b;
        return sum;

}
//위 사용자 정의 함수는 12-05
int cha(int a, int b) {
    int sub;
    printf("\n [뺄셈 cha() 함수 영역] \n\n");
    sub = a - b;
    return sub;

}
//위 사용자 정의 함수는 12-05
int sum(const int sale[Y][P])
{
    int a, b;
    int hap = 0;

    for (a = 0; a < Y; a++)
        for (b = 0; b < P; b++)
            hap += sale[a][b];

    return hap;
}
//위 사용자 정의 함수는 10 -16
int b_s(int list[], int n, int input)
{
    int low = 0;
    int high = n - 1;
    int md, cnt = 1;

    while (low <= high)
    {
        printf("%d회 이진 탐색 : [%d %d] \n", cnt, low, high);

        md = (low + high) / 2;

        if (input == list[md])
            return md;
    }
}

```

```

        else if (input > list[md])
            low = md + 1;

        else
            high = md - 1;

        cnt++;
    }
    return -1;
}

// 위 사용자 정의 함수는 10-12
int gob(int ta)
{
    int mul;

    mul = ta * ta;
    printf("\n사용자 정의 함수 gob()호출\n");
    printf("산출 > su[3] 배열 요소의 제곱 : %d * %d = %d\n", ta, ta, mul);

    return mul;
}

int sv(int score[], int size) {
    int cnt, total = 0;

    printf("\n사용자 정의 함수로 총점 계산\n");

    for (cnt = 1; cnt < size; cnt++)
        total += score[cnt];

    return total;
}

//위 사용자 정의 함수는 10-8
int main() {
    int score[5] = { 87,92,89,98,78 };
    int cnt;

    printf("score[5] = { 87,92,89,98,78 } 배열 요소 출력");

    for (cnt = 0; cnt < 5; cnt++)
        printf("score[%d] = [%d]\n", cnt, score[cnt]);
}

```

```

return 0;
//line 18 :10-1

int score[3];
int cnt, sum = 0;
float avg;

for (cnt = 0; cnt < DT; cnt++) {
    printf("과목%d 점수 : __\b\b\b", cnt + 1);
    scanf_s("%d", &score[cnt]);
}
for (cnt = 0; cnt < DT; cnt++)
    sum += score[cnt];

avg = (float)sum / DT;

printf("총점 : %5d\n", sum);
printf("평균 : %5.2f\n", avg);
return 0;
//line 33: 10-2

int ex[PT];
int cnt;

printf("1~500 사이 숫자를 랜덤 출력\n");
srand((unsigned)time(NULL));

for (cnt = 0; cnt < PT; cnt++)
    ex[cnt] = 1 + (rand() % 500);
for (cnt = 0; cnt < PT; cnt++)
    printf("ex[%d] = %d\n", cnt, ex[cnt]);

return 0;
//line 50 10-3

int point[] = { 23,35,2,9,75,96,82,16,48,85,33,67,95,37 };
int count;

printf("point[] = { 23,35,2,9,75,96,82,16,48,85,33,67,95,37 }\n\n");
printf("제시한 배열 요소의 개수 출력\n");

```



```

count = sizeof(point) / sizeof(point[0]);

printf("point[] 배열 요소의 개수: %d개\n ", count);
return 0;
//line 62 10-4

char yn;
int choice, cnt;
int car[CP] = { 0 };

do
{
    printf("자동차 모델을 선택하겠습니까? (Y / N): ");
    scanf_s("%c", &yn, sizeof(yn));

    if (yn == 'N' || yn == 'n')
        break;
    else if (yn == 'Y' || yn == 'y')
    {
        printf("자동차 모델 번호 : 1 2 3 4 5 6 7 8 9 10\n");
        printf("자동차 예약 현황 :");

        for (cnt = 0; cnt < CP; cnt++)
            printf("%2d", car[cnt]);
        printf("\n");
        re_input: //goto 문 레이블
        printf("\n자동차 모델 선택(1~10) : ");
        scanf_s("%d", &choice);

        if (choice < 1 || choice > 10)
        {
            printf("잘못 입력했습니다!\n");
            printf("모델 번호를 다시 입력하세요.\n");
            goto re_input;
        }
        else
        {
            if (car[choice - 1] == 0)
            {
                car[choice - 1] = 1;
            }
        }
    }
}

```

```

        printf("예약 완료! \n");
    }
    else
    {
        printf("[No.%d] 모델은 이미 예약되었습니다.\n",
choice);

        printf("모델 번호를 다시 선택하세요.\n");
        goto re_input;
    }
}
else
{
    printf("알파벳 입력 오류!\n");
    printf("알파벳은 대소 문자 구별 없이 Y 또는 N만 허용합니다.
\n");

    printf("알파벳을 다시 입력하시오.\n\n");
}
} while (1);

printf("자동차 모델 [No.%d] 예약 완료 후 프로그램 종료\n");
return 0;
//line 118 10-5

char chs1[6] = { 'a','p','p','l','e','\0' };
char chs2[6] = "apple";
char chs3[6] = "apple";

printf("배열 크기를 6과 5로 설정한 출력\n");
printf("1.배열 크기 6: 요소를 중괄호로 초기화 . . .[%s]\n",chs1);
printf("2.배열 크기 6: 한꺼번에 문자열 초기화 . . .[%s]\n",chs2);
printf("3.배열 크기 6: 한꺼번에 문자열 초기화 . . .[%s]\n",chs3);
return 0;
//line 129 10-6

char str[20];
int cnt, length = 0;

printf("문> 문자열을 20바이트 이내로 입력하시오 : ");
scanf_s("%s", str, sizeof(str));

```

```

for (cnt = 0; str[cnt] != '\0'; cnt++)
    length++;
printf("\n");
printf("답 > 입력한 문자열 \"%s\"의 길이 : %d바이트 \n ", str, length);
return 0;
//line 142 10-7

```

```

int score[DT] = { 88,96,75,81,99 };
int sum = 0;
double avg;

```

```

printf("상수> #define DT 5\n");
printf("배열 > score[DT] = { 88,96,75,81,99 }; \n");

```

```

sum = sv(score, DT);
avg = sum / 5.0;

```

```

printf("반환 > 5개 과목의 총점: [%5d]\n", sum);
printf("산출 > 5개 과목의 평균: [%5.2f]\n", avg);
return 0;
//10 -8

```

```

int su[DT] = { 1,2,3,4,5 };
int result;

```

```

printf("상수> #define DT 5\n");
printf("배열> su[DT] = {1,2,3,4,5}\n");
printf("전달 > su[3] = { %d} --> 배열 요소 복사본 전달\n", su[3]);

```

```

result = gob(su[3]);

```

```

printf("반환 > gob(su[3])호출 결과 반환 : [%2d]\n", result);
return 0;
//10 -9

```

```

int s_list[AL] = { 5,4,1,2,3 };
int a, b ,temp, sort;

```

```

printf("초기 > 배열 요소 : 5 4 1 3 2");
printf("\n[for 문으로 선택 정렬 수행]\n");

```

```

for (a = 0; a < AL - 1; a++)
{
    sort = a;

    for (b = a + 1; b < AL; b++)
        if (s_list[b] < s_list[sort])
            sort = b;

    temp = s_list[a];
    s_list[a] = s_list[sort];
    s_list[sort] = temp;
}
printf("결과 > 배열 요소:");

for (a = 0; a < AL; a++)
    printf("%d", s_list[a]);

return 0;
// 10-10

int s_list[AL] = { 5,4,1,3,2,7,6 };
int input, cnt;
char yn;

printf("초기 > 배열 요소: 5 4 1 3 2 7 6\n");

while (1)
{
    printf("숫자 입력(1~7):");
    scanf_s("%d", &input);
    printf("\n");

    if (input < 1 || input > 7)
    {
        printf("값 입력 오류!\n");
        printf("다시 입력하세요.\n");
        continue;
    }
    for(cnt = 0; cnt <AL; cnt++)
        if (s_list[cnt] == input)
            {

```

```

        printf("순차 탐색 횟수 : %d회\n", cnt + 1);
        printf("탐색 성공 인덱스 : [%d]\n", cnt);
    }

re_ch:

    printf("프로그램을 다시 시작할까요? (Y / N) : ");
    scanf_s(" %c", &yn, sizeof(yn));

    if (yn == 'N' || yn == 'n')
        break;
    else if (yn == 'Y' || yn == 'y')
        continue;
    else
    {
        printf("알파벳 입력 오류!\n");
        printf("알파벳을 다시 입력하세요.\n");
        goto re_ch;
    }
}

return 0;
//10-11
int sno[AL] = { 3,8,15,24,35,57,68,85 };
int input, search;
char yn;

do
{
    printf("\n초기> 배열 요소 : 3 8 15 24 35 57 68 85\n");
    printf("숫자 입력 : ");
    scanf_s("%d", &input);

    search = b_s(sno, AL, input);
    printf("탐색 성공 인덱스 : [%2d]\n",search);

re_input:

    printf("프로그램을 다시 실행하겠습니까? (Y/N) : ");
    scanf_s("%c", &yn, sizeof(yn));

    if (yn == 'N' || yn == 'n')

```

```

        break;
    else if (yn == 'Y' || yn == 'y')
        continue;
    else
    {
        printf("\n알파벳 입력 오류!\n");
        printf("알파벳을 다시 입력하세요.\n");
        goto re_input;
    }
} while (1);
return 0;
// 10 -12

```

```

int cnt, sno = 203012101, total = 0;
double avg;
int bd[R][C] = {
    {135,150,148,160,153},
    {115,120,123,121,112},
    {132,129,128,133,130}};

```

```

printf(">> 고혈압 판정 기준 : 수축기 혈압 140 초과 \n");
for (cnt = 0; cnt < R; cnt++)
{
    total = bd[cnt][0] + bd[cnt][1] + bd[cnt][2] + bd[cnt][3] + bd[cnt][4];
    avg = total / 5.0;

    printf("\n > 환자 #%d의 수축기 평균 혈압 및 판정 \n ", cnt + 1);
    printf("번호   : %d\n", sno++);
    printf("혈압   : %.2f\n", avg);

    if (avg > 140)
        printf("판정   : 고혈압\n");

    else
        printf("판정   : 정상 \n");
}
return 0;
//10 -13

```

```

int row, col, total = 0;
double abg;

```

```

char name[N];
int st[R][C];

for (row = 0; row < R; row++)
{
    printf("> 학생 %d의 성명과 3개 과목 점수 입력\n", row + 1);
    printf("%d.성명 :", row + 1);

    scanf_s("%s", name, sizeof(name));

    for (col = 0; col < C; col++)
    {
        printf("과목%d 점수 : ", col + 1);
        scanf_s("%d ", &st[row][col]);
    }
    printf("\n");

    total = st[row][0] + st[row][1] + st[row][2];
    avg = total / 3.0;

    printf("\%s\`의 성적 현황 \n", name);
    printf("총점 : [%5d]\n", total);
    printf("평균 : [%.2f]\n\n", avg);
}
printf("전체 %d명의 성적 처리 후 프로그램 종료\n", row);
return 0;
// 10-14

```

```

int tic[R][C];
int row, col, p, g;

printf("게임 실행 횟수 제한: 총 9회\n");
grid();

for (row = 0; row < R; row++)
    for (col = 0; col < C; col++)
        tic[row][col] = ' ';

printf("게임 요령 -> 홀수 횟수와 짝수 횟수로 상대방 구분\n");
printf("[사용자1] : 홀수 횟수 사용\n");
printf("[사용자2] : 짝수 횟수 사용\n");

```

```

printf("[허용 범위] : 0 ~ 2까지만 허용!\n\n");

for (p = 0; p < 9; p++)
{
    re_play:
        printf("%d번째 입력 : ", p + 1);
        scanf_s("%d %d", &row, &col);

        if (row < 0 || row > 2 || col < 0 || col > 2)
        {
            printf("입력 오류!\n");
            printf("다시 입력하세요.\n\n");
            goto re_play;
        }
        tic[row][col] = (p % 2 == 0) ? '0' : 'X';
    }
printf("\n 게임 결과 \n");

for (g = 0; g < R; g++)
{
    printf("|---|---|---|\n");
    printf("| %c | %c | %c |\n", tic[g][0], tic[g][1], tic[g][2]);
}
printf("|---|---|---|\n\n");
printf("총 %d회를 실행하였으므로 프로그램 종료\n", p);
return 0;
// 10 - 15

int row, col, total = 0;
int sale[Y][P] = {
    {2025,353,127,83,883},
    {2026,387,133,89,907},
    {2027,439,135,92,965}};

printf("컴퓨터\ 모니터\ 키보드\ 마우스\ 소프트웨어\n");

for (row = 0; row < Y; row++)
{
    for (col = 0; col < P; col++)
        printf("%d\t", sale[row][col]);
    printf("\n");
}

```



```
}
```

```
total = sum[sale];
```

```
printf("3년간 총 매출액 : %7d백만 원\n ", total);
```

```
return 0;
```

```
// 10 -16
```

```
int row, col, class, total = 0;
```

```
double avg;
```

```
int st[R][C];
```

```
re:
```

```
printf("학급수 입력(1~3) :");
```

```
scanf_s("%d", &class);
```

```
if (class < 1 || class > 3)
```

```
{
```

```
    printf("입력값 오류!\n");
```

```
    printf("학급수를 다시 입력하세요.\n\n");
```

```
    goto re;
```

```
}
```

```
printf("%d개 학급에 대한 성적 처리 시작\n", class);
```

```
for (cnt = 1; cnt <= class; cnt++)
```

```
{
```

```
    printf("학급%d의 학생%d의 3개 과목 점수 입력\n", cnt, row + 1);
```

```
    for (col = 0; col < C; col++)
```

```
    {
```

```
        printf("과목%d의 점수 : ", cnt, col + 1);
```

```
        scanf_s("%d", &st[row][col]);
```

```
    }
```

```
    total = st[row][0] + st[row][1] + st[row][2];
```

```
    avg = total / 3.0;
```

```
    printf("총점 : [%5d]\n", total);
```

```
    printf("평균 : [%.2f]\n\n", avg);
```

```
}
```

```
printf("전체 %d개 학급의 성적 처리 프로그램 종료!\n", class);
```

```
return 0;
```

```
//10 -17
```

```

int pg, row, col;
int index[P][R][C] = {
    {{1,2,3,4,5},
     {6,7,8,9,10},
     {11,12,13,14,15}},
    {{16,17,18,19,20},
     {21,22,23,24,25},
     {26,27,28,29,30}}};
for (pg = 0; pg < P; pg++)
{
    printf("\n[%d면] 출력\n", pg + 1);

    for (row = 0; row < R; row++)
    {
        printf("%d행 출력 ...\n", row +1);
        for (col = 0; col < C; col++)
        {
            printf("%2d",index[pg][row][col]);
            printf("[%d][%d][%d]",pg ,row, col);
        }
        printf("\n");
    }
}
return 0;
//10 - 18

```

```

int pg, row, col, cnt = 0, total = 0;
double avg;
int oil[P][R][C];
char pro[K][M] = { "현대","쌍용" };
char car[CR][CC] = { "승용차", "SUV" };

printf("%s와 %s의 제조사별 주유 금액 산출 시작\n\n",pro[0],pro[1]);

for (pg = 0; pg < P; pg++)
{
    for (row = 0; row < R; row++)
    {
        printf("제조사 : %s, 3차원 배열 중 [면]\n",pro[pg]);
        printf("1.차종 스타일 : %s ,3차원 배열 중 [행]\n",car[row]);
    }
}

```

```
printf("2. 연차별 주유 금액(단위:백만 원),3차원 배열 중  
[열]\n\n");
```

```
for (col = 0; col < C; col++)  
{  
    printf("%d년차 주유 금액 : ", col + 1);  
    scanf_s("%d", &oil[pg][row][col]);  
}  
total = oil[pg][row][0] + oil[pg][row][1] + oil[pg][row][2];  
avg = total / 3.0;  
  
printf("\n3년간 총 주유 금액: [%6d]백만 원\n",total);  
printf("연간 평균 주유 금액: [%6.2f]백만 원 \n\n",avg);  
}  
}  
return 0;  
//10 -19 10장 예제 마무리.
```

```
int i_num = 3;  
char ch = 'A';  
float f_num = 21.5;  
  
printf("자료형별 주소를 10진수로 출력\n");  
printf("1. int형 변수의 주소 : %u\n",&i_num);  
printf("2. char형 변수의 주소 : %u\n",&ch);  
printf("3. float형 변수의 주소: %u\n", &f_num);
```

```
printf("\n자료형별 주소를 16진수로 출력\n");  
printf("1. int형 변수의 주소: %X\n", &i_num);  
printf("2. char형 변수의 주소 : %X\n",&ch);  
printf("3. float형 변수의 주소 : %X\n",&f_num);  
return 0;  
//11 -01
```

```
int a = 0;  
int* pa = NULL;
```

```
pa = &a;
```

```
float b = 3.75;  
float* pb = &b;
```

```

printf("일반 변수와 포인터 초기화\n");
printf("int a = 8;\n");
printf("int *pa = NULL;\n");
printf("pa = &a;\n");
printf("float b = 3.75\n");
printf("float *pb = &b;\n");

printf("\n일반 변수의 값과 포인터의 주소값 출력\n");
printf("1.일반 변수 a의 값 : %u\n",a);
printf("2.포인터 pa의 주소값 : %u\n",pa);
printf("3.일반 변수 &a의 주소값 : %u\n",&a);

printf("\n일반 변수를 사용하여 주소값 출력\n");
printf("1. 포인터 pb의 주소값 : %u\n",pb);
printf("2. 일반 변수 &b의 주소값: %u\n",&b);
printf("3. NULL의 주소값 : %u\n", NULL);
return 0;
//11-02

```

```

int a = 300;
int* pa = NULL;

pa = &a;
char yn;

do
{
    printf("\n포인터 주소값과 일반 변수의 데이터값 출력\n");
    printf("1.포인터 pa의 주소값 : %u\n",pa);
    printf("2.일반 변수 a의 데이터값: %u\n",a);

    printf("\n* 연산자와 & 연산자 사용해 출력\n");
    printf("1.주소 연산자 &를 사용한 주소값: %u\n",&a);
    printf("2. 간접 참조 연산자 *pa의 데이터값 %u\n",*pa);

re_play:
    printf("프로그램을 다시 시작하겠습니까?(Y/N) : ");
    scanf_s("%c", &yn, sizeof(yn));

    switch (yn)

```

```

    {
    case 'Y':case 'y':
        continue;
        break;

    case 'N':case 'n':
        printf("프로그램을 종료합니다.\n");
        return 0;
    default:
        printf("\n%c는(은) 허용하지 않는 알파벳입니다.\n",yn);
        printf("알파벳을 다시 입력하세요.\n");
        goto re_play;
        break;
    }
} while (1);
// 11-03

```

```

int a = 30;
int* p = NULL;
printf("a = 30, p = &a일 때\n");

```

```

p = &a;
printf("1.포인터 p의 주소값 : %u\n",p);
printf("2.포인터가 가리키는 *p의 데이터값 : %u\n", *p);

```

```

printf("\n*p = 50 일때\n");

```

```

*p = 50;
printf("1.포인터 p의 주소값 : %u\n",p);
printf("2.포인터가 가리키는 *p의 데이터값 : %u\n",a);
return 0;
// 11 -04

```

```

int a = 103;
int* pa = NULL;

```

```

pa = &a;

```

```

printf("int a =103\n");
printf("int *pa =NULL;\n");

```

```
printf("pa = &a;\n");
```

```
if (pa != NULL)
```

```
{
```

```
    printf("pa는 유효한 포인터\n");
```

```
    printf("1.포인터 pa의 주소값 : %u\n",pa);
```

```
    printf("2.포인터가 가리키는 *pa의 데이터값 : %u\n",*pa);
```

```
}
```

```
else
```

```
{
```

```
    printf("소스 코드 08행을 /* pa = &a; */처럼 주석 처리한 경우\n");
```

```
    printf("pa는 유효하지 않은 포인터!\n");
```

```
}
```

```
return 0;
```

```
//11-05
```

```
int a;
```

```
double* pd = NULL;
```

```
pd = &a;
```

```
*pd = 13.8;
```

```
if (pd != NULL)
```

```
{
```

```
    printf("pd는 유효한 포인터\n");
```

```
    printf("1.포인터 pd의 주소값 : %u\n",pd);
```

```
    printf("2.포인터가 가리키는 *pd의 데이터값 : %u\n",*pd);
```

```
}
```

```
else
```

```
    printf("pd는 유효하지 않은 포인터\n");
```

```
return 0;
```

```
//11-06
```

```
char c;
```

```
char* pc;
```

```
pc = &c;
```

```
int i;
```

```
int* pi;
```

```
pi = &i;
```

```

double d;
double* pd;
pd = &d;

printf("1.포인터 증감 연산자 사용 전 주소값\n");
printf("char형 포인터 주소값 : %d\n",pc);
printf("int형 포인터 주소값 : %d\n",pi);
printf("double형 포인터 주소값: %d\n",pd);

*pc++;
*pi++;
*pd++;

printf("\n2.포인터 *p++ 연산자 수행 후 주소값\n");
printf("char형 포인터 주소값 : %d\n",pc);
printf("int형 포인터 주소값 : %d\n",pi);
printf("double형 포인터 주소값 : %d\n",pd);
return 0;
//11-07

```

```

double d = 3.3058;
double* pd;

pd = &d;

printf("1.포인터 pd의 증감 연산 전 주소값과 데이터값\n");
printf("포인터 pd의 현재 주소값 : %u\n",pd);
printf("포인터 pd의 현재 데이터값 : %f\n",d);

```

```

(*pd)++;

printf("\n2.포인터 (*pd)++ 증감 연산 후 주소값과 데이터값\n");
printf("포인터 pd의 증감 연산 후 주소값 : %u\n",pd);
printf("포인터 pd의 증감 연산 후 데이터값 : %u\n",d);
return 0;
//11-08

```

```

int a[] = { 100,200,300 };
int* pa;

```

```
pa = a;
```

```
printf("int a[] = {100, 200, 300} 일 때 포인터 증감 연산\n");
printf("\n1.포인터 *++pa의 증감 연산을 수행한 주소값과 데이터값\n");
printf("포인터 현재 주소값 : %d\n",pa);
printf("현재 포인터 주소의 데이터값 : %d\n",*a);
printf("포인터 *++pa 수행 후 데이터값 : %d\n", *++pa);
printf("포인터 *++pa 수행 후 주소값 : %d\n",pa);
```

```
printf("\n2.포인터 *--pa의 증감 연산을 수행한 주소값과 데이터값\n");
printf("포인터 현재 주소값 : %d\n",pa);
printf("포인터 *--pa 수행 후 데이터값 : %d\n", *--pa);
printf("포인터 *--pa 수행 후 주소값 : %d\n", pa);
```

```
printf("\n3.포인터 *++pa의 증감 연산을 수행한 주소값과 데이터값\n");
printf("포인터 현재 주소값 : %d\n", pa);
printf("포인터 *++pa 수행 후 데이터값 : %d\n", ++*pa);
printf("포인터 *++pa 수행 후 주소값 : %d\n", pa);
return 0;
//11- 09
```

```
int a[] = { 100,200,300,400,500 };
int cnt;
```

```
printf("int a[] = {100,200,300,400,500}일 때\n");
printf("\n1. 배열 요소의 주소값 출력\n");
```

```
for(cnt = 0; cnt < R; cnt++)
    printf("%d번째 배열 요소 a[%d] = a[%d]\n",cnt + 1, cnt,&a[cnt]);
```

```
printf("\n2. 주소값과 배열 요소의 값 출력\n");
printf("배열의 현재 주소값 : a는 %u\n",a);
printf("첫 번째 배열 요소의 값 : *a는 %u\n",*a);
```

```
printf("\n증감 연산 후 주소값 : a + 1은 %u\n",a + 1);
printf("증감 연산 후 배열 요소의 값 : *(a + 1)은 %u\n",(a + 1));
```

```
printf("\n배열의 현재 주소값 : a는 %u\n",a);
printf("\n증감 연산 후 주소값 : a+ 2는 %u\n", a + 2);
printf("증감 연산 후 배열 요소의 값 : *(a + 2)은 %u\n", (a + 2));
```



```

return 0;
//11-10

int a[R] = { 100,200,300,400,500 };
int* pa, cnt, hap = 0;

pa = a;
printf("int a[] = {100,200,300,400,500}일 때 \n\n");

for (cnt = 0; cnt < R; cnt++)
{
    hap += *pa++;
    printf("%d번째 배열 요소 : %d", cnt + 1 , a[cnt]);
    printf("포인터 주소 : %d\n", pa);
}
printf("\n%d개의 배열 요소 합계 : %d\n", cnt,hap);
return 0;
// 11-11 11장 끝

```

```

char sa[] = "Spacezone";
const char* ps = "Spacezone";
int cnt;

printf("배열 선언 char sa[]= \"Spacezone\"\n");
printf("포인터 선언 char *ps= \"Spacezone\"\n");
printf("\n 배열 주소값: %d\n", sa);
printf("\n 포인터 주소값: %d\n", *ps);

printf("\n[배열사용] \t [포인터 사용]\n");
for (cnt = 0; cnt < 9; cnt++) {
    printf("sa[%d] : %c\t\t", cnt, sa[cnt]);
    printf("*(ps+%d) : %c\n", cnt, *(ps + cnt));
}
return 0;
//12-01

```

```

int cnt, size, choice;
const char* task[] = {
    "Desk",
    "Computer",
    "Books",

```

```

        "USB",
        "EXIT"
};

printf("포인터 배열의 요소 선택(종료: 5)\n");

size = sizeof(task) / sizeof(task[0]);
for (cnt = 0; cnt < size; cnt++) printf("%d.%s\n", cnt + 1, task[cnt]);

while (1) {
    printf("\n숫자 입력 (1 ~ 5): ");
    scanf_s("%d", &choice);
    if (choice < 1 || choice > 5) {
        printf("입력 숫자 오류!\n");
        continue;
    }
    else if (choice == 5) break;
    else
        printf("선택한 단어는 \"%s\"입니다.\n", task[choice - 1]);
}
return 0;
//12-02

int a[5] = { 10,20,30,40,50 };
int(*ap)[5];
int cnt;

printf("배열 선언: int a[5] = { 10,20,30,40,50 }\n ");
printf("배열 포인터 선언: int (*ap)[5]\n");

ap = &a;
printf("\n 배열 포인터 ap의 주소 : %d\n", ap);
printf("배열 요소 출력 : a[5] = { 10,20,30,40,50 }\n\n");

for (cnt = 0; cnt < 5; cnt++) {
    printf("%d번째 요소 : %d\n", cnt + 1, (*ap)[cnt]);

    return 0;
}
//12-03

```

```

int a[2][4] = {
    {10,20,30,0},
    {40,50,60,0}
};
int(*four)[4];
int row, col;

four = a;
printf("2차원 배열 요소와 행에 대한 합계 출력\n");

for (row = 0; row < 2; row++)
    for (col = 0; col < 3; col++)
        four[row][3] += four[row][col];

for (row = 0; row < 2; row++)
{
    printf("%d번째 행 배열 요소 출력\n", row + 1);
    for (col = 0; col < 3; col++)
    {
        printf("[%d][%d]", row, col);
        printf("%d\t\n", four[row][col]);

    }
    printf("배열 요소 합계: [%d]\n\n", four[row][3]);
}
return 0;
//12-04

int result;
int(*fp)(int, int);

printf("100과 200의 덧셈 연산과 뺄셈 연산\n");

fp = hap;
result = fp(100, 200);

printf("덧셈 연산 결과 : %d\n", result);

fp = cha;
result = fp(100, 200);

```

```
printf("뽕셈 연산 결과 : %d \n", result);
return 0;
//12-05
```

```
int choice, result, a, b;
int (*fp[4])(int, int) = { hap,cha,gob,na }; // 함수 포인터의 배열 선언
```

```
while (1)
{
    menu();
re:
    printf("선택 (0~4):");
    scanf_s("%d", &choice);
    if (choice < 0 || choice>4)
    {
        printf("\n 입력 오류 \n");
        printf("다시 선택하세요. \n");

        goto re;
    }
    else if (choice == 4) break;
    else
    {
        printf("정수 2개 입력 (예> 10 20) : ");
        scanf_s("%d %d", &a, &b);
        result = fp[choice](a, b); // 함수 포인터 배열로 선택한 함수 호

        printf("연산결과: %d\n", result);
    }
    printf("프로그램 수행 완료\n");
}
printf("\프로그래밍 종료\n");
return 0;
//12-06
```

출

```
const char* season[] = { "Spring","Summer","Fall","Winter" };
char c = 'F';
const char* pc, ** ppc;
int cnt;
```

```

pc = &c;
ppc = &pc;

printf("일반 변수 c : %c \n", c);
printf("포인터 변수 *pc : %c \n", *pc);
printf("이중 포인터 **ppc : %c \n", **ppc);
ppc = season;

printf("\n");
printf("char *season[] = {\"Spring\\\", \"Summer\\\", \"Fall\\\", \"Winter\\\"}\n");
for (cnt = 0; cnt < 4; cnt++)

    printf("%d번째 배열 요소인 *(ppc + %d) 출력 : %s\n", cnt + 1, cnt,
*(ppc + cnt));
return 0;
//12-07

char* pc;
printf("나의 생활 신조와 인생 좌우명 \n\n");

creed(&pc); // 포인터 변수 pc 의 시작 주소를 전달하여 함수 호출(pc 값은 변경)
printf("생활 신조 : %s\n", pc);
target(&pc);
printf("인생 좌우명 : %s\n", pc);
return 0;
//12-08

int a = 100;
char b = 'F';
void* vp;

printf("1.캐스트 연산자를 사용하는 경우 \n");

vp = &a;
printf("변수 a의 값이 *(int*)vp로 출력 : %d\n", *(int*)vp);

vp = &b;
printf("변수 b의 값이 *(cahr*)vp로 출력 : %c\n", *(char*)vp);

printf("\n2. 캐스트 연산자를 사용하지 않는 경우 \n");
printf("변수 a의 값이 *vp로 출력 : Illegal Indirection 오류 \n");

```

```
printf("변수 b의 값이 *vp로 출력 : Illegal Indirection 오류 \n");  
return 0;  
//12-09
```

```
char str[80];  
printf("문자열 입력 : ");
```

```
gets_s(str);  
bubble(str, strlen(str));  
printf("문자열을 버블 정렬한 결과 : %s \n", str);  
return 0;  
//12-10
```

```
char str[80];
```

```
printf("문자열 입력 : ");  
gets_s(str);
```

```
selection(str, strlen(str));
```

```
printf("문자열을 선택 정렬한 결과 : %s \n", str);  
return 0;  
//12-11
```

```
char str[80];  
char key = 'k';  
int check;
```

```
printf("문자열 입력 : ");  
gets_s(str); // 키보드로부터 문자열 입력받을 배열 선언
```

```
check = sequential(str, strlen(str), key); // 인수 3개를 전달하여 호출  
if (check < 1) printf("입력한 문자열에는 알파벳 소문자 \"%k\"가 없습니다. \n");  
else  
{  
    printf("알파벳 소문자 \"%k\"를 찾았습니다. \n");  
    printf("문자열의 %d번째에 있습니다.\n", check + 1);  
}  
return 0;
```

```
//12-12
```

```
char str[80];  
char key = 'k';  
int check;
```

```
printf("문자열 입력 : ");  
gets_s(str);
```

```
check = binary(str, strlen(str), key);  
if (check < 1) {  
    printf("입력한 문자열에는 알파벳 소문자 \"%k\"가 없습니다. \n");  
}  
else  
{  
    printf("알파벳 소문자 \"%k\"를 찾았습니다. \n");  
    printf("문자열의 %d번째에 있습니다.\n", check + 1);  
}  
return 0;  
//12-13 12장 마무리
```

```
char str[3] = "SPEED";  
printf("문자열 출력: %s\n", str);  
return 0;  
//13-01
```

```
char str[8] = "SPEED";  
printf("문자열 출력: %s\n", str);  
return 0;  
//13-02
```

```
char* str = "Hello~";  
  
printf("문자열 출력 : %s\n", str);  
str = "Have a goot time!";  
printf("문자열 출력 : %s\n", str);  
return 0;  
//13-03
```

```
char* str = "Hello~";
```

```
printf("문자열 출력 : %s\n", str);
str = "Have a good time!";
printf("문자열 출력 : %s\n", str);
return 0;
//13-04
```

```
int cnt;
```

```
printf("수강 신청 과목 현황\n");
char* dept[] = { "운영체제", "빅데이터 분석", "사물인터넷", "소프트웨어공학" };
for (cnt = 0; cnt < 4; cnt++)
    printf("과목%d: %s\n", cnt + 1, dept[cnt]);
```

```
return 0;
//13-05
```

```
char munja;
```

```
printf("문자 입력:");
munja = getchar();
```

```
printf("문자출력:");
putchar(munja);
return 0;
//13-06
```

```
int munja;
while (1) {
    if ((munja = getchar()) != EOF)
        putchar(munja);
    else
        break;
}
return 0;
//13-07
```

```
int munja;
```

```
printf("1.getch() 함수 사용\n");
printf("문자 입력:");
munja = _getch();
```



```
printf("\n 문자 입력:");
_putchar(munja);
```

```
printf("\n\n 2._getche() 함수 사용\n");
printf("문자 입력:");
munja = _getche();
```

```
printf("\n문자 출력:");
_putchar(munja);
return 0;
//13-08
```

```
unsigned char op;
```

```
printf("문자 입력:");
op = _getch();
```

```
while (op != '\r') {
    printf("\n입력한 문자: %c\n", op);
    printf("아스키코드값 : %d\n", op);
    op = _getch();
}
printf("프로그램 종료\n");
return 0;
//13-09
```

```
int op;
printf("문자 입력 : ");
op = getchar();
if (isalnum(op))
    printf("입력한 %c는 (은) 숫자 또는 알파벳입니다.", op);
else
    printf("입력한 %c는 (은) 숫자 또는 알파벳이 아닙니다.\n", op);

return 0;
//13-10
```

```
int op;
```

```
printf("문자 입력:");
op = getchar();
```

```

if (isalnum(op))
{
    printf("입력한 %c는 (은) 숫자 또는 알파벳입니다.", op);
    if (isupper(op)) printf("입력한 %c는(은) 알파벳 대문자입니다.\n", op);
    else printf("입력한 %c는(은) 알파벳 소문자 입니다.\n", op);
}
else
    printf("입력한 %c는 (은) 숫자 또는 알파벳이 아닙니다.\n", op);

return 0;
//13-11

int index = 0, cnt = 0;
char str[] = "korea \t space \n";
printf("charf str[] = \"korea \\t space \\n\"으로 선언\n");

while (str[index])
{
    if (iscntrl(str[index])) cnt++;
    index++;
}
printf("제어문자는 모두 %d개 존재합니다. \n", cnt);
return 0;
//13-12

int op;

printf("문자입력: ");

op = getchar();
if (isalpha(op))
{
    printf("입력한 문자 %c은(는) 알파벳이 맞습니다. \n", op);
    if (islower(op))
    {
        op = toupper(op);
        printf("알파벳을 대문자 %c로 변환합니다.\n", op);
    }
    else

```

```

        {
            op = tolower(op);
            printf("알파벳을 소문자  %c로 변환합니다.\n", op);
        }
    }
else

```

```

    printf("입력한 %c는(은) 알파벳이 아닙니다. \n", op);

```

```

return 0;
//13-13

```

```

char name[20];
char dept[30];

```

```

printf("성명: ");
gets_s(name, 20);

```

```

printf("학과: ");
gets_s(dept, 30);

```

```

printf("\n입력 사항 출력 \n");

```

```

printf("성명 : ");
puts(name);

```

```

printf("학과 : ");
puts(dept);
return 0;
//13-14

```

```

char dept[20];

```

```

printf("다음 세 과목 중에서 전공을 선택하세요. \n");
printf("웹 언어 / 정보 보안 / 네트워크 \n");

```

```

printf("\n1.gets_s() 함수 사용: ");
gets_s(dept, sizeof(dept));

```

```

printf("선택한 전공 : ");
puts(dept);

```

```
printf("\n2.scanf_s() 함수 사용 : ");  
scanf_s("%s", dept, sizeof(20));
```

```
printf("선택한 전공 : ");  
puts(dept);  
return 0;  
//13-15
```

```
char* chs;  
int str1;  
double str2;
```

```
chs = "12345";  
str1 = atoi(chs);  
printf("\n12345\> 정수형 int로 변환 : %d\n", str1);  
chs = "12a34";  
str1 = atol(chs);  
printf("\n12a34\> 정수형 long로 변환 : %d\n", str1);
```

```
chs = "12.345";  
str2 = atof(chs);  
printf("\n12.345\> 실수형 float로 변환 : %f\n", str2);
```

```
chs = "a12.34";  
str2 = atof(chs);  
printf("\na12.34\> 실수형 float로 변환 : %f\n", str2);  
return 0;  
//13-16
```

```
char* str = "C Program";  
char data[20] = "";
```

```
printf("주어진 문자열 : %s\n", str);  
printf("문자열의 길이 : %d\n", strlen(str));
```

```
strcpy_s(data, sizeof(data), str);  
printf("\n 복사한 문자열 : %s\n\n", data);
```

```
strncpy_s(data, sizeof(data), str, ST);  
printf("5개 복사한 문자열 : %s \n", data);
```

```
printf("문자열의 길이 : %d \n", strlen(data));  
return 0;  
//13-17
```

```
char s1[20];  
char s2[20];  
int result;
```

```
printf("s1 문자열 입력 : ");  
gets_s(s1, sizeof(s1));
```

```
printf("s2 문자열 입력 : ");  
gets_s(s2, sizeof(s2));
```

```
result = strcmp(s1, s2);  
printf("\nstrcmp() 함수로 문자열 비교 결과\n");  
if (result > 0)  
{  
    printf("s1 > s2 비교 결과값 : %d\n", result);  
    printf("문자열 비교 : s1이 s2 문자열 뒤에 있음\n");  
}  
else if (result == 0)  
{  
    printf("s1 > s2 비교 결과값 : %d\n", result);  
    printf("문자열 비교 : s1이 s2 문자열이 같음\n");  
}  
else  
{  
    printf("s1 > s2 비교 결과값 : %d\n", result);  
    printf("문자열 비교 : s1이 s2 문자열 앞에 있음\n");  
}  
return 0;  
//13-18
```

```
char s1[15] = "Space";  
char s2[5] = "Zone";
```

```
printf("s1 문자열 : %s\n", s1);  
printf("s2 문자열 : %s\n", s2);
```

```
strcat_s(s1, sizeof(s1), s2);
```

```
printf("\n 문자열 전체 결합: %s\n", s1);
```

```
strncat_s(s1, sizeof(s1), s2, ST);
```

```
printf("문자열 일부 결합 : %s\n", s1);
```

```
return 0;
```

```
//13-19
```

```
char s[] = "school";
```

```
char* p;
```

```
char chk;
```

```
printf("문자열 : %s\n", s);
```

```
printf("검색할 문자 입력:");
```

```
scanf_s("%c", &chk, sizeof(s));
```

```
p = strchr(s, chk);
```

```
if (p != NULL)
```

```
    printf("문자열 %s에는 존재합니다.\n", s, chk);
```

```
else
```

```
    printf("문자열 %s에는 존재하지 않습니다.\n", s, chk);
```

```
return 0;
```

```
//13-20
```

```
char s[] = "C Professional Project Team";
```

```
printf("문자열 : %s\n", s);
```

```
printf("문자열에서 ' 문자가 포함된 부분 검색\n");
```

```
printf("\nstrchr() 함수 사용 :%s\n\n", strchr(s, 'P'));
```

```
printf("문자열에서 \"Pro\" 문자열이 포함된 부분 검색\n");
```

```
printf("strstr() 함수 사용 :%s\n", strstr(s, "Pro"));
```

```
return 0;
```

```
//13-21
```

```
char str[] = "Have a Good Time";
```

```
printf("문자열 : %s\n", str);
```

```
_strrev(str);
```

```
printf("\n 문자열을 역순으로 변환\n");
printf("_strrev() 함수 사용 : %s\n", str);
```

```
_strrev(str);
```

```
printf("문자열을 원래대로 변환");
printf("_strrev() 함수 사용 : %s\n", str);
return 0;
//13-22
```

```
char str[] = "Have a Good Time";
printf("문자열 : %s\n", str);
```

```
_strupr_s(str);
```

```
printf("\n 알파벳을 모두 대문자로 변환\n");
printf("_strupr_s() 함수 사용 : %s\n", str);
```

```
_strlwr_s(str);
```

```
printf("알파벳을 모두 소문자로 변환");
printf("_strlwr_s() 함수 사용 : %s\n", str);
return 0;
//13-23
```

```
char str[] = "Let's do-our_best! today";
char div[] = "-_!";
char* tok;
char* context;
int cnt = 1;
```

```
printf("주어진 문자열 : %s\n", str);
printf("구분자 : 공백 (\\" \"), 하이픈(-), 언더바(_), 느낌표(!)\n\n");
```

```
tok = strtok_s(str, div, &context);
```

```
while (tok != NULL) {
    printf("%d번째 토큰 : %s\n", cnt, tok);
    tok = strtok_s(NULL, div, &context);
    cnt++;
}
```

```
}  
return 0;  
//13-24
```

```
int cnt;  
char s[4][7] = {  
    "Spring",  
    "Summer",  
    "Fall",  
    "Winter"  
};
```

```
printf("사계절을 for문으로 출력\n");
```

```
for (cnt = 0; cnt < 4; cnt++)  
    printf("%02d번째 계절: %s\n", cnt + 1, s[cnt]);
```

```
return 0;  
//13-25
```

```
int cnt;  
char s[3][20];
```

```
printf("3개 수강 신청과목 입력\n");
```

```
for (cnt = 0; cnt < 3; cnt++) {  
    printf("%02d번째 과목 입력:", cnt + 1);  
    scanf_s("%s", s[cnt], 20);  
}
```

```
printf("\n3개 수강 신청 과목 출력\n");
```

```
for (cnt = 0; cnt < 3; cnt++)  
    printf("%02d번째 과목 출력:[%s]\n", cnt + 1, s[cnt]);
```

```
return 0;  
//13-26
```

```
while (1) {  
    int cnt, index = 0;  
    int result;  
    char word[30];
```



```

char memter[5][2][20] = {
    {"space","123456"},
    {"web2me","000000"},
    {"power","555555"},
    {"quen33","888888"},
    {"zzang","111111"}
};
printf("ID 입력:");
scanf_s("%s", word, 30);
for (cnt = 0; cnt < 5; cnt++) {
    result = strcmp(memter[cnt][0], word);

    if (result == 0) {
        printf("ID [%s]의", word);
        printf("PASSWORD는 %s입니다.\n", memter[cnt][1]);
        return 0;
    }
    index++;
}
printf("입력한 ID%s는 찾을 수 없습니다.\n", word);
printf("ID를 다시 입력하세요.\n");
}
//13-27 13장 마무리
}

```