

=포인터 연습문제=

2. i와 j는 int형이고, p와 q는 int형 포인터라고 하자.

다음 배정수식중 잘못된 것은 어느것인가?

```
p = &i  p = &&*i  i = (int)p  q = &p
*q = &j  i = (*&)j  i = *&*j    i = *p++ + *q
```

8. p가 포인터라면 *p++과 (*p)++는 다르다. 다음 프로그램을 실행하여 그 결과를 보고 두 수식이 어떻게 다른지 설명하여라.

```
#include <stdio.h>
int main(void){
    char string[] = "abc";
    char *p;
    int i;
    p = string;
    for(i = 0; i < 3; ++i)
        printf("%c\\n", *p++);
    printf("string = %s\\n", string);
    for(i = 0; i < 3; ++i)
        printf("%c\\n", (*p)++);
    printf("string = %s\\n", string);
    return 0;
}
```

11.같은 내용의 문자열 상수를 여러 번 사용하면 이들이 같은 메모리에 하나만 저장되는지 아니면 사용될때마다 다른 메모리에 따로 저장되는지를 다음 프로그램으로 확인해보아라.

```
#include <stdio.h>
int main(void){
    char *p = "C is fun.", *q = "C is fun.";
    if(p == q)
        printf("같은 메모리에 하나만 저장됩니다.");
    else
        printf("다른 메모리에 따로 저장됩니다.");
}
```

12.다음 프로그램에서 func()함수 원형의 매개변수 형은 int[][3]인 반면 func()함수의 정의헤더에서 매개변수 형은 int(*a)[3]이다. 컴파일할 때, 형이 다르다는 오류가 발생하는가? 컴파일 된다면 실행결과는 어떻게 되는가?

```
#include <stdio.h>
void func(int[][3]);
int main(){
    int a[4][3] = {{0,1,2}, {10,11,22}, {20,21,22}, {30,31,32}};
    func(a);
    return 0;
}
void func(int (*a)[3]){
    printf("func() : %d %d %d\\n", a[0][0], a[1][1], a[2][2]);
}
```

이제 func()함수의 정의 헤더에서 매개변수 형을 int *a[3]으로 수정하면 오류가 난다. 이유는?

15. 다음 프로그램의 실행결과를 설명하여라.

```

#include <stdio.h>
#define N 5
void change_it(int []);
int main(void){
    int a[N] = {1,2,3,4,5}, i;
    printf("change_it 호출 전\n");
    for(i = 0; i < N ; i++)
        printf("a[%d] = %d\n", i, a[i]);
    change_it(a);
    printf("change_it 호출 후\n");
    for(i = 0; i < N ; i++)
        printf("a[%d] = %d\n", i, a[i]);
    return 0;
}
void change_it(int a[]){
    int b[N] = {6,7,8,9,10};
    a = b;
}

```

20. C컴파일러는 함수 이름을 포인터로 취급한다. 함수 포인터를 역참조하면 함수 자신이 되기 때문에 또 함수 포인터라고 생각할 수 있다. 또한 함수 포인터 뒤에 괄호를 붙이면 함수 호출을 의미한다. 다음 프로그램은 이에 관한 예제이다. 올바르게 실행되는지 확인

```

#include <stdio.h>
void f(void);
void g(void);
void h(void);
int main(void){
    (*f());
    return 0;
}
void f(void){
    printf("f()가 호출되었습니다.\n");
    (((*g)))();
}
void g(void){
    printf("g()가 호출되었습니다.\n");
    (*( (*h)))();
}
void h(void){
    printf("h()가 호출되었습니다.\n");
}

```

=구조체 연습문제=

17. 다음 프로그램은 어떤 값을 출력하겠는가? 출력결과에 대해 설명하여라.

```

#include <stdio.h>
enum day {SUN, MON, TUE, WED, THU, FRI, SAT};
enum name {kim, lee, park, jung, hong = 266};
int main(void){

```

```

printf("sizeof(enum day) = %d\n", sizeof(enum day));
printf("sizeof(enum name) = %d\n", sizeof(enum name));
return 0;
}

```

18. 다음 프로그램에 오류가 있는가? 컴파일해보아라. 오류메세지가 출력되는가?
어떤 컴파일러는 오류메세지를 출력할 것이고, 어떤 것은 정상적으로 컴파일할 것이다.

```

#include <stdio.h>
enum day {SUN, MON, TUE, WED, THU, FRI, SAT};
enum name {kim, lee, park, jung, hong = 266};
int main(void){
    enum day d;
    d = kim;
    return 0;
}

```

19. 다음 프로그램은 어떻게 출력될 것인가? 실행해 보고 원하지 않은 결과가 출력된다면 올바른 결과가 출력되도록 프로그램을 수정하여라.

```

#include <stdio.h>
enum day {SUN, MON, TUE, WED, THU, FRI, SAT};
int main(){
    enum day d = SUN;
    printf("오늘은 %d입니다.\n", d);
    return 0;
}

```

20. 이전 달을 리턴하는 previous_month() 함수를 작성하여라. 이 함수는 다음과 같이 시작해야 한다.

```

enum month {Jan = 1, Feb, Mar, Apr, May, Jun, Jul,
            Aug, Sep, Oct, Nov, Dec};
typedef enum month month;

```

Jan이 함수의 인자로 전달되면 Dec이 리턴되어야 한다. 또한 달의 이름을 출력하는 printf_month() 함수도 작성하여라. 프로그램을 작성하여 작성한 함수들을 확인하여라.

=매크로 연습문제=

5. 인자를 갖는 매크로를 사용하는 프로그램은 디버깅하기가 어렵다. 다음 다음 프로그램은 인자로 넘어오는 문자열을 출력하기 위해 PRN(x)라는 매크로를 정의하였다. 올바르게 수행되는지 확인해 보아라.

```

#include <stdio.h>
#define PRN(x) printf("x\n");
int main(void){
    PRN(Hello from main());
    return 0;
}

```

무엇이 문제인지 확인하고 올바르게 동작하도록 수정하여라.

8. forever() 매크로가 다음과 같이 정의되어 있다고 하자.

```

#define forever(x) forever(forever(x))

```

이때 forever(x)를 호출하면 어떻게 확장되는지 확인해보아라.

11. 반지름이 r인 원의 넓이는 $r^2 \cdot \pi$ 이다. 원의 넓이를 계산하는 매크로와

inline함수는 다음과 같이 정의할 수 있다.

```
#define PI 3.14
#define AREA(r) ((r) * (r) * PI)
inline float area_f(float r) {return r * r * PI;}
```

AREA()와 area_f()각각을 사용하여 반지름이 1부터 10까지 1씩 증가할 때의 원의 넓이 표를 만들어보자. 이를 위한 프로그램은 다음과 같이 만들 수 있을 것이다.

```
#include <stdio.h>
#define PI 3.14
#define AREA(r) ((r) * (r) * PI)
inline float area_f(float r) {return r * r * PI;}
int main(){
    float a;
    int i = 1;
    printf("By macroWnr    areaWn");
    while(i <= 10){
        printf("%d : ", i);
        a = AREA(i++);
        printf("%.2fWn", a);
    }
    i = 1;
    printf("By inline functionWnr    areaWn");
    while(i <= 10){
        printf("%d : ", i);
        a = area_f(i++);
        printf("%.2fWn", a);
    }
    return 0;
}
```

12. 밑변이 w이고 높이가 h인 삼각형의 넓이는 $w \cdot h / 2$ 이다. 삼각형 넓이를 구하는 inline 함수는 다음과 같이 정의할 수 있다.

```
static inline float tri_area_f(float w, float h)
{return w*h/2;}
```

이 inline 함수를 매크로로 정의하면 다음과 같이 할 수 있을 것이다.

```
#define TRIE_AREA(w, h) ((w) * (h) / 2)
```

밑변이 1이고 높이가 1인 삼각형의 넓이를 함수와 매크로를 사용하여 구하는 프로그램은 다음과 같이 만들수 있을 것이다.

```
#include <stdio.h>
static inline float tri_area_f(float w, float h) {return w*h/2;}
#define TRIE_AREA(w, h) ((w) * (h) / 2)
int main(void){
    float a;
    printf("By inline functionWn");
    a = tri_area_f(1,1);
    printf("The area is %fWn", a);
    printf("By macroWn");
    a = TRIE_AREA(1,1);
    printf("The area is %fWn", a);
}
```

```

    return 0;
}

```

이 프로그램을 실행시켜보아라. 같은 결과가 나오는가? 다르다면 이유를 설명하여라.

15. 문자열화 연산자 #은 매크로로 전달되는 인자를 큰따옴표로 묶는다. 매크로의 인자가 이미 큰따옴표로 묶여 있다면 어떻게 되는지 다음 프로그램을 수행하여 알아보아라.

```

#include <stdio.h>
#define test(x) s = #x
int main(void){
    char *s;
    test("Hello!");
    printf("%s\n", s);
    return 0;
}

```

20. 다음 프로그램은 0부터 9까지의 제곱을 출력하기 위해 작성한 프로그램이다. 실행하여보고 그 결과에 대해 설명하여라. 올바르게 실행되도록 수정하여라.

```

#include <stdio.h>
#define SQR(x) ((x) * (x))
int main(){
    for(int i = 0; i < 10; ){
        printf("%d * %d = ", i, i);
        printf("%d\n", SQR(i++));
    }
    return 0;
}

```

파일 입출력 예제 및 연습문제

예제 11.6 입력받은 파일을 뒤에서부터 거꾸로 출력하는 프로그램

```

#include <stdio.h>
#define MAXSTRING 100
int main(void){
    char fname[MAXSTRING];
    int c;
    FILE *ifp;
    fprintf(stderr, "\n입력 파일 : ");
    scanf("%s", fname);
    ifp = fopen(fname, "r");
    fseek(ifp, 0, SEEK_END);
    if(ftell(ifp) == 0){
        printf("Empty file...\n");
        return 0;
    }
    fseek(ifp, -1, SEEK_CUR);
    while(1){
        c = getc(ifp);
        putchar(c);
    }
}

```

```

        if(ftell(ifp) == 1){
            break;
        }
        fseek(ifp, -2, SEEK_CUR);
    }
    fclose(ifp);
    return 0;
}

```

예제 11.7 학생 이름과 성적을 입력받아 그 내용을 이진파일로 저장하는 예제이다.

```

#include <stdio.h>
typedef struct student{
    int id;
    char name[10];
    int grade[3];
    int sum;
    float avg;
}student;
int main(int argc, char **argv){
    FILE *ofp;
    int id, check;
    student st = {0, "", {0}, 0, 0.0};
    if(argc != 2){
        fprintf(stderr, "사용법 : %s out_fileWn", argv[0]);
        exit(1);
    }
    if((ofp = fopen(argv[1], "wb")) == NULL){
        fprintf(stderr, "오류 %s파일을 열 수 없습니다.Wn", argv[1]);
        exit(1);
    }
    fprintf(stderr, "성적을 입력하세요.Wn");
    fprintf(stderr, "입력 형식 : 이름 수학성적 영어성적 과학성적Wn");

    id = 1;
    check = scanf("%s %d %d %d", st.name, &st.grade[0],
                  &st.grade[1], &st.grade[2]);
    while(check != EOF){
        st.id = id++;
        fwrite(&st, sizeof(student), 1, ofp);
        check = scanf("%s %d %d %d", st.name, &st.grade[0],
                      &st.grade[1], &st.grade[2]);
    }
    fclose(ofp);
    return 0;
}

```

예제 11.8 프로그램 11.7에서 만든 이진 파일을 읽는 프로그램이다.

```

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

```

```

typedef struct student{
    int id;
    char name[10];
    int grade[3];
    int sum;
    float avg;
}student;
int main(int argc, char **argv){
    FILE *ifp;
    int check;
    student st = {0, "", {0}, 0, 0.0};
    if(argc != 2){
        fprintf(stderr, "사용법 : %s out_fileWn", argv[0]);
        exit(1);
    }
    if((ofp = fopen(argv[1], "rb")) == NULL){
        fprintf(stderr, "오류 %s파일을 열 수 없습니다.Wn", argv[1]);
        exit(1);
    }
    fprintf(stderr, "번호 이름      수학성적 영어성적 과학성적Wn");
    check = fread(&st, sizeof(student), 1, ifp);
    while(check){
        fprintf("%3d %-9s %3d %3d %3dWn", st.id, st.name, st.grade[0],
            st.grade[1], st.grade[2]);
        check = fread(&st, sizeof(student), 1, ifp);
    }
    fclose(ifp);
    return 0;
}

```

예제 11.9 이진 파일로 저장된 성적 파일을 읽어, 전체출력, 학생별 출력, 성적 추가, 성적 수정, 성적처리등을 하는 프로그램이다.

```

#include <stdio.h>
#include <stdlib.h>
typedef struct student{
    int id;
    char name[10];
    int grade[3];
    float avg;
} student;
int selectNum(void);
int main(int argc, char **argv){
    FILE *fp;
    int check, id, sel;
    student st = {0, "", {0}, 0.0};
    if(argc!=2){
        fprintf(stderr, "사용법 : %s grade_fileWn", argv[0]);
        exit(1);
    }
}

```

```

if((fp = fopen(argv[1], "rb+")) == NULL){
    fprintf(stderr, "오류 : %s 파일을 열 수 없습니다.Wn", argv[1]);
    exit(1);
}
sel = selectNum();
while(sel){
    switch(sel){
        case 1:
            rewind(fp);
            fprintf(stderr, "번호 이름 수학 영어 과학 평균Wn");
            check = fread(&st, sizeof(student), 1, fp);
            while(check){
                fprintf(stderr, "%3d %-9s %3d %3d %3d %6.2fWn",
                    st.id, st.name, st.grade[0], st.grade[1], st.grade[2], st.avg);
                check = fread(&st, sizeof(student), 1, fp);
            }
            break;
        case 2:
            fprintf(stderr, "학생 id : ");
            scanf("%d", &id);
            fseek(fp, sizeof(student)*(id-1), SEEK_SET);
            check = fread(&st, sizeof(student), 1, fp);
            fprintf(stderr, "번호 : %3d 이름 : %9s Wn", st.id, st.name);
            fprintf(stderr, " 수학 : %3d, 영어 : %3d, 과학 : %3d",
                st.grade[0], st.grade[1], st.grade[2]);
            fprintf(stderr, " 평균 : %.2fWn", st.avg);
            break;
        case 3 :
            fprintf(stderr, "이름과 성적을 입력하세요 : ");
            check = scanf("%s %d %d %d", st.name,
                &st.grade[0], &st.grade[1], &st.grade[2]);
            fseek(fp, 0, SEEK_END);
            st.id = ftell(fp) / sizeof(student) + 1;
            st.avg = (st.grade[0] + st.grade[1] + st.grade[2]) / 3.0;
            fwrite(&st, sizeof(student), 1, fp);
            getchar();
            break;
        case 4 :
            fprintf(stderr, "학생 id : ");
            scanf("%d", &id);
            fseek(fp, sizeof(student) * (id - 1), SEEK_SET);
            check = fread(&st, sizeof(student), 1, fp);
            fprintf(stderr, "%s 학생 성적을 입력하세요 : ", st.name);
            check = scanf("%s %d %d %d", st.name,
                &st.grade[0], &st.grade[1], &st.grade[2]);
            st.avg = (st.grade[0] + st.grade[1] + st.grade[2]) / 3.0;
            fseek(fp, -sizeof(student), SEEK_CUR);
            fwrite(&st, sizeof(student), 1, fp);
    }
}

```



```

        break;
    case 5 :
        rewind(fp);
        check = fread(&st, sizeof(student), 1, fp);
        while(check){
            st.avg = (st.grade[0] + st.grade[1] + st.grade[2])/3.0;
            fseek(fp, -sizeof(student), SEEK_CUR);
            fwrite(&st, sizeof(student), 1, fp);
            check = fread(&st, sizeof(student), 1, fp);
        }
        fprintf(stderr, "성적 처리 완료\n");
        break;
    default :
        fprintf(stderr, "잘못된 번호입니다. 다른 번호 선택하세요.\n");
    }
    sel = selectNum();
}
fprintf(stderr, "프로그램을 종료합니다.\n");
fclose(fp);
return 0;
}

```

```

int selectNum(void){
    int num;
    fprintf(stderr, "\n\n원하는 번호를 입력하세요.\n");
    fprintf(stderr, "1 : 전체 성적 출력\n");
    fprintf(stderr, "2 : 학생별 출력\n");
    fprintf(stderr, "3 : 성적 추가\n");
    fprintf(stderr, "4 : 성적 수정\n");
    fprintf(stderr, "5 : 성적 처리\n");
    fprintf(stderr, "0 : 종료\n");
    fprintf(stderr, "input > ");
    scanf("%d", &num);
    return num;
}

```

연습문제 13 sscanf()는 fscanf()의 문자열 버전이다. 하지만 연속적으로 사용했을 때 읽는 위치가 다르다. 파일을 열면 파일 위치 지시자는 다음에 어디서부터 읽어야 하는지를 유지한다. 하지만 문자열 버전인 sscanf()에는 이와 같은 매커니즘이 없다. 다음코드를 포함하는 프로그램을 작성하여 tmp1과 tmp2에 어떤 내용이 출력되는지 확인하고 그 내용에 대해 설명하여라.

```

char c, s[] = "abc", *p = s;
int i;
FILE *ofp1, *ofp2;
ofp1 = fopen("tmp1", "w");
ofp2 = fopen("tmp2", "w");
for(i = 0; i < 3; i++){
    sscanf(s, "%c", c);
    fprintf(ofp1, "%c", c);
}

```

```

for(i = 0; i < 3; i++){
    sscanf(p++, "%c", c);
    fprintf(ofp2, "%c", c);
}

```

=정적 외부변수=

예제 12.4

```

//main.c
#include <stdio.h>
int hours, minutes;
void convertM2HM(int);
int main(){
    extern int hours, minutes;
    int inputM;

    printf("분을 입력하세요 : ");
    scanf("%d", &inputM);

    convertM2HM(inputM);
    printf("%d분은 %d시간 %d분 입니다.Wn", inputM, hours, minutes);
    return 0;
}

//convert.c
void convertM2HM(int min){
    extern int hours, minutes;
    hours = min/ 60;
    minutes = min % 60;
    return;
}

```

예제 12.5

```

//main.c
#include <stdio.h>
int access(int);
int change_passwd(int);
int main(){
    int num, new, select = 1;
    while (select){
        printf("1 : Login, 2 : Change password, 0 : Exit");
        printf("Select number : ");
        scanf("%d", &select);
        switch(select){
            case 0 :
                printf("Bye...Wn");
                break;
            case 1 :
                printf("Password : ");
                scanf("%d", &num);
                if(access(num))

```

```

        printf("Wrong password\n");
    else
        printf("Welcome....");
    break;
case 2 :
    printf("New password : ");
    scanf("%d", &new);
    if(change_passwd(new))
        printf("Login first...\n");
    else
        printf("Password has been changed\n");
    break;
default :
    printf("Input 0 to 2...\n");
}
}
return 0;
}
//password.c
#include <stdbool.h>
static int passwd = 78;
static bool is_login = false;
static void set_login(bool ok){
    is_login = ok;
}
int access(int num){
    if(num != passwd){
        set_login(false);
        return -1;
    }
    set_login(true);
    return 0;
}
int change_passwd(int new){
    if(!is_login)
        return -1;
    passwd = new;
    return 0;
}

```

=스택=

예제 12.6 스택을 이용한 프로그램

```

//stack.h
#include <stdio.h>
#include <stdbool.h>
#define MAX_LEN 100
#define EMPTY -1
#define FULL (MAX_LEN - 1)

```

```

typedef struct stack{
    char s[MAX_LEN];
    int top;
}stack;
void reset(stack *stk);
void push(char c, stack *stk);
char pop(stack *stk);
char top(const stack *stk);
bool empty(const stack *stk);
bool full(const stack *stk);
//main.c
#include "stack.h"
int main(){
    char str[] = "Stack Test!";
    int i;
    stack s;
    reset(&s);

    printf("문자열 : %s\n", str);
    //스택에 문자열 push
    for(i = 0; str[i] != '\0'; ++i){
        if(!full(&s)){
            push(str[i], &s);
        }
    }
    printf("역 문자열 : ");
    //스택에서 문자열 pop
    while(!empty(&s)){
        putchar(pop(&s));
    }
    putchar('\n');
    return 0;
}
//stack.c
#include "stack.h"
void reset(stack *stk){
    stk->top = EMPTY;
}
void push(char c, stack *stk){
    stk->top++;
    stk->s[stk->top] = c;
}
char pop(stack *stk){
    return (stk->s[stk->top--]);
}
char top(const stack *stk){
    return (stk->s[stk->top]);
}

```

```

bool empty(const stack *stk){
    return (stk -> top == EMPTY);
}
bool full(const stack *stk){
    return (stk -> top == FULL);
}

```

=자기참조 구조체=

예제 12.12

```

#include <stdio.h>
#include <stdlib.h>
typedef struct list{
    char ch;
    struct list *next;
}list;
int main(void){
    list *p;
    p = (list *)malloc(sizeof(list));
    p -> ch = 'a';
    p -> next = (list *)malloc(sizeof(list));
    p -> next -> ch = 'a';
    p -> next -> next= (list *)malloc(sizeof(list));
    p -> next -> next -> ch = 'a';
    p -> next -> next -> next = p;

    printf("a : '%c', b : '%c', c : '%c'\n",
           p->ch, p->next->ch,p->next->next->ch);
    return 0;
}

```