

PUZZLE

BOOK

차례

1. Operators

-1- 1. Basic Arithmetic Operators
-1- 2. Assignment Operators
-1- 3. Logic and Increment Operators
-1- 4. Bitwise Operators
-1- 5. Relational and Conditional Operators
-1- 6. Operator Precedence and Evaluation

2. Basic Types

-2- 1. Character, String, and Integer Types
-2- 2. Integral and Floating Point Casts
-2- 3. More Casts

3. Control Flow

-3- 1. if Statement
-3- 2. while and for Statements
-3- 3. Statement Nesting
-3- 4. wswitch, break, and continue

4. Programming Style

-4- 1. Choose the Right Condition
-4- 2. Choose the Right Construct

5. Storage Classes

-5- 1. Blocks
-5- 2. Functions
-5- 3. More Functions
-5- 4. Files

차례

차례

6. Pointers and Arrays

-6- 1. Simple Pointer and Array
-6- 2. Array of Pointers
-6- 3. Multidimensional Array
-6- 4. Pointer stew

7. Structures

-7- 1. Simple Structure, Nested Structure
-7- 2. Array of Structures
-7- 3. Array of Pointers to Structures

8. Preprocessor

-8- 1. The Preprocessor Knows Little C
-8- 2. Caution Pays

차례

Operator 1 : Basic Arithmetic Operators

What does the following program print?

```
#include <stdio.h>

void main()
{
    int x;

    x = -3 + 4 * 5 - 6;
    printf("%d\n",x);
    x = 3 + 4 % 5 - 6;
    printf("%d\n",x);
    x = -3 * 4 % -6 / 5;
    printf("%d\n",x);
    x = (7 + 6) % 5 / 2;
    printf("%d\n",x);
}
```

Operator 2 : Assignment Operators

What does the following program print?

```
#include <stdio.h>
#define PRINTX printf("%d\n",x)

void main()
{
    int x=2 , y , z;

    x *= 3 + 2;
    PRINTX;
    x *= y = z = 4;
    PRINTX;
    x = y == z;
    PRINTX;
    x == (y = z);
    PRINTX;
}
```

Operators 3 : Logic and Increment Operators

What does the following program print?

```
#include <stdio.h>
#define PRINT(int) printf("%d\\", int)

void main()
{
    int x, y, z;

    x = 2; y = 1; z = 0;
    x = x && y || z;
    PRINT(x);
    PRINT(x || !y && z);

    x = y = 1;
    z = x++ - 1;
    PRINT(x);
    PRINT(z);

    z += -x++ + ++y;
    PRINT(x);
    PRINT(z);

    z = x / ++x;
    PRINT(z);
}
```

Operators 4 : Bitwise Operators

What does the following program print?

```
#include <stdio.h>
#define PRINT(int) printf("#int " = "%d\\",int);

void main()
{
    int x, y, z;

    x=03; y=02 ;z=01;
    PRINT(x | y & z);
    PRINT(x | y & ~z);
    PRINT(x ^ y & ~z);
    PRINT(x & y && z);

    x=1; y=-1;
    PRINT(!x | x);
    PRINT(~x | x);
    PRINT(x ^ x);
    x<=3;
    PRINT(x);
    y<=3;
    PRINT(y);
    y>=3;
    PRINT(y);
}
```

Operators 5 : Relational and Conditional Operators

What does the following program print?

```
#include <stdio.h>
#define PRINT(int) printf("#int "= %d\n",int);

void main()
{
    int x=1, y=1, z=1;
    x += y += z;
    PRINT(x<y ? y : x);

    PRINT(x<y ? x++ : y++);
    PRINT(x); PRINT(y);

    PRINT(z += x<y ? x++ : y++);
    PRINT(y); PRINT(z);

    x=3; y=z=4;
    PRINT( (z >= y >= x) ? 1 : 0);
    PRINT(z >= y && y >= x);
}
```

Operators 6 : Operator Precedence and Evaluation

What does the following program print?

```
#include <stdio.h>
#define PRINT3(x,y,z) printf("#x"= %d\n"#y"= %d\n"#z"= %d\n",x,y,z)

void main()
{
    int x, y, z;
    x=y=z=1;
    ++x || ++y && z++;PRINT3(x,y,z);

    x=y=z=1;
    ++x && ++y || ++z;PRINT3(x,y,z);

    x=y=z=1;
    ++x && ++y && ++z;PRINT3(x,y,z);

    x=y=z=-1;
    ++x && ++y || ++z;PRINT3(x,y,z);

    x=y=z=-1;
    ++x || ++y && z;PRINT3(x,y,z);

    x=y=z=-1;
    ++x && ++y && ++z;PRINT3(x,y,z);
}
```

Basic Types 1 : Character, String, and Integer Types

What does the following program print?

```
#include <stdio.h>
#define PRINT(format,x) printf("#x"= "%"#format"W",x);
int integer=5;
char character='5';
char * string="5";

void main()
{
    PRINT(d, string); PRINT(d, character); PRINT(d, integer);
    PRINT(s, string); PRINT(c, character); PRINT(c, integer=53);
    PRINT(d,('5'>5));
    {
        int x = -2;
        unsigned int ux=-2;
        PRINT(o, x); PRINT(o, ux);
        PRINT(d, x/2); PRINT(d, ux/2);
        PRINT(o, x>>1); PRINT(o, ux>>1);
        PRINT(d, x>>1); PRINT(d, ux>>1);
    }
}
```

Basic Types 2 : Integer and Floating Point Casts

What does the following program print?

```
#include <stdio.h>
#define PR(x) printf("#x"= "%.8gW",(double)x)
#define NL putchar('W')
#define PRINT4(x1,x2,x3,x4) PR(x1);PR(x2);PR(x3);PR(x4);NL

void main()
{
    double d;
    float f;
    long l;
    int i;
    i=l=f=d=100/3; PRINT4(i,l,f,d);
    d=f=l=i=100/3; PRINT4(i,l,f,d);
    i=l=f=d=100/3.; PRINT4(i,l,f,d);
    d=f=l=i=(float)100/3; PRINT4(i,l,f,d);
    i=l=f=d=(double)(100000/3); PRINT4(i,l,f,d);
    d=f=l=i=100000/3; PRINT4(i,l,f,d);
}
```

Basic Types 3 : More Casts

What does the following program print?

```
#include <stdio.h>
#define PR(x) printf("#x"= %gW", (double) (x))
#define NL putchar('W')
#define PRINT1(x1) PR(x1);NL
#define PRINT2(x1,x2) PR(x1);PRINT1(x2)

void main()
{
    double d= 3.2, x;
    int i= 2,y;

    x= (y= d/i) *2; PRINT2(x,y);
    y= (x= d/i) *2; PRINT2(x,y);
    y= d*(x= 2.5/d); PRINT1(y);
    x= d*(y= ((int)2.9+ 1.1)/d); PRINT2(x,y);
}
```

이다음부터 사용할 사용자 정의 헤더 파일 내용

defs.h로 저장

```
#include <stdio.h>
#define PR(fmt,val) printf("#val " = %" #fmt "W", (val))
#define NL putchar('W')
#define PRINT1(f,x1) PR(f,x1),NL
#define PRINT2(f,x1,x2) PR(f,x1),PRINT1(f,x2)
#define PRINT3(f,x1,x2,x3) PR(f,x1),PRINT2(f,x2,x3)
#define PRINT4(f,x1,x2,x3,x4) PR(f,x1),PRINT3(f,x2,x3,x4)
```

Control Flow 1 : If Statement

what does the following program print?

```
#include "defs.h"

void main()
{
    int x,y=1,z;
    if(y!=0){ x=5; }
    PRINT 1(d,x);
    if(y==0){ x=3; }
    else{ x=5; }
    PRINT 1(d,x);
    x=1;
    if(y<0) if(y>0){ x=3; }
    else{ x=5; }
    PRINT 1(d,x);
    if(z=y<0){ x=3; }
    else if(y==0){ x=5; }
    else{ x=7; }
    PRINT 2(d,x,z);
    if(z=(y==0)){ x=5; }
    x=3;
    PRINT 2(d,x,z);
    if(x=z=y){ ; } x=3;
    PRINT 2(d,x,z);
}
```

Control Flow2 : While and for Statements

What does the following program print?

```
#include "defs.h"

void main()
{
    int x,y,z;
    x=y=0;
    while(y<10){
        ++y;
        x+=y;
    } PRINT 2(d,x,y);

    x=y=0;
    while(y<10){
        x+=++y;
    } PRINT 2(d,x,y);

    y=1;
    while(y<10){
        x=y++;z=++y;
    } PRINT 3(d,x,y,z);

    for(y=1;y<10;y++){ x=y; } PRINT 2(d,x,y);
    for(y=1;(x=y)<10;y++){ PRINT 2(d,x,y);
    for(x=0,y=1000;y>1;x++,y/=10){ PRINT 2(d,x,y); }
}
```


Control Flow3 : Statement Nesting

What does the following program print?

```
#include "defs.h"
#define ENUF 3
#define EOS '\0'
#define NEXT(i) input[i++]
#define FALSE 0
#define TRUE 1
char input[] = "PI=3.14159,approximately";

void main()
{
    char c;
    int done, high, i, in, low;

    i=low=in=high=0;
    while(c= NEXT (i) != EOS)
        if(c<'0'){ low++; }
        else if(c>'9'){ high++; }
        else{ in++; }
    PRINT 3(d,low,in,high);

    i=low=in=high=0;
    done=FALSE;
    while((c= NEXT (i))!= EOS && !done)
        if(c<'0'){ low++; }
        else if(c>'9'){ high++; }
        else{ in++; }
        if(low>= ENUF || high>= ENUF || in>= ENUF){ done=TRUE; }
    PRINT 3(d,low,in,high);
```

}

```
i=low=in=high=0;
done=FALSE;
while((c= NEXT (i))!= EOS && !done)
    if(c<'0') done= (++low== ENUF? TRUE : FALSE);
    else if(c>'9') done= (++high== ENUF ? TRUE : FALSE);
    else done= (++in== ENUF ? TRUE : FALSE);
PRINT 3(d,low,in,high);
```

Control Flow 4 : switch, break, and continue

what does the following program print?

```
#include "defs.h"
char input[] = "SSSWILTECH1WWW1WWWALLMP1";

void main()
{
    int i,c;
    for(i=2;(c=input[i])!='\0';i++){
        switch(c){
            case 'a':putchar('i');continue;
            case '1': break;
            case 1: while((c=input[++i])!='W' && c!='0');
            case 9: putchar('S');
            case 'E':
            case 'L': continue;
            default : putchar(c);continue;
        }
        putchar(' ');
    }
    putchar('\n');
}
```

//Programming Style 은 뭐 따로 어떠한 내용이 있는것이 아니라 그냥 이런식으로 짤다
하는 흐름이더라구요 책에 있길래 그냥 타이핑은 해줬습니다~

Programming Style 1 : Choose the Right Condition

Improve the following program fragments through reorganization

```
while(A){
    if(B) continue;
    C;
}
do {
    if(!A) continue;
    else B;
    C;
} while(A);
```

```
if(A){
    if(B){
        if(C){
            D;
        }
        else;
    }
    else;
}
else{
    if(B){
        if(C){
            E;
        }
        else{
```

```
        F;
    }
}
else;
}
while((c= getchar()) != '\0'){
    if(c== ' ') continue;
    if(c== '\n') continue;
    if(c< '0') return(OT HER);
    if(c<= '9') return(DIGIT);
    if(c< 'a') return(OT HER);
    if(c<= 'z') return(ALPHA);
    return(OT HER);
}
return(EOL);
```

Programming Style 2 : Choose the Right Construct

Improve the following program fragments through reorganization

```
done= i= 0;
while(i< MAXI && ~done){
    if((x/= 2)> 1){i++; continue;}
    done++;
}
plusflg= zeroflg= negflg= 0;
if(a> 0) ++ plusflg;
if(a= 0) ++ zeroflg;
else if(!plusflg) ++ negflg;
if(A){B; return;}
if(C){D; return;}
if(E){F; return;}
G; return;
i= 0;
while((c= getchar())!= EOF){
    if(c!= '\n' || c!= '\0'){s[i++] = c; continue;}
    if(c= '\n') break;
    if(c= '\0') c= ' ';
    s[i++] = c;
}
if(x!= 0){
    if(j> k) y= j/x;
    else y= k/x;
}
else{
    if(j> k) y= j/NEARZERO;
    else y= k/NEARZERO; }
```

Storage Classes 1 : Blocks

What does the following program print?

```
#include "defs.h"

int i= 0;

void main()
{
    auto int i= 1;
    PRINT 1(d,i);
    {
        int i= 2;
        PRINT 1(d,i);
        {
            i+= 1;
            PRINT 1(d,i);
        }
        PRINT 1(d,i);
    }
    PRINT 1(d,i);
}
```

Storage Classes 2 : Functions

What does the following program print?

```
#include "defs.h"
#define LOW 0
#define HIGH 5
#define CHANGE 2
void workover(int);
int reset(int);
int i=LOW;

void main()
{
    auto int i=HIGH;
    reset(i/2);PRINT 1(d,i);
    reset(i=i/2);PRINT 1(d,i);
    i=reset(i/2);PRINT 1(d,i);
    workover(i); PRINT 1(d,i);
}

void workover(int i)
{
    i=(i%i)*((i*i)/(2*i)+4);
    PRINT 1(d,i);
}

int reset(int i)
{
    i=i<=CHANGE ? HIGH : LOW;
    return (i);
}
```

Storage Classes 3 : More Functions

What does the following program print?

```
#include "defs.h"

int i= 1;
int reset();
int next(int);
int last(int);
int nEw(int);

void main()
{
    auto int i,j;
    i= reset();
    for(j= 1;j<= 3;j+ ){
        PRINT 2(d,i,j);
        PRINT 1(d,next(i));
        PRINT 1(d,last(i));
        PRINT 1(d,nEw(i+j));
    }
}

int reset(void)
{
    return (i);
}

int next(int j)
{
    return (j= i+ );
}

int last(int j)
{
```

```
static int i= 10;
return (j= i- );
}

int nEw(int i)
{
    auto int j= 10;
    return(i= j+ = i);
}
```

Storage Classes 4 : Files

what does the following program print?

```
#include "defs.h"
int i= 1;

void main()
{
    auto int i,j;
    i= reset();
    for(j= 1;j<= 3; j++ ){
        PRINT 2(d,i,j);
        PRINT 1(d,next());
        PRINT 1(d,last());
        PRINT 1(d,nEw(i+j));
    }
}

//new file
extern int i;
int reset(void)
{
    return (i);
}

//new file
static int i= 10;

int next(void)
{
    return (i+= 1);
}
```

```
int last(void)
{
    return (i-= 1);
}

int nEw(int i)
{
    static int j= 5;
    return(i= j+= i);
}
```

Pointer and Arrays 1 : Simple Pointer and Array

what does the following program print?

```
#include "defs.h"
int a[]={0,1,2,3,4};

void main()
{
    int i, *p;
    for(i=0;i<=4;i++){PR(d,a[i]);}
    NL;
    for(p=&a[0]; p<=&a[4];p++){ PR(d,*p); }
    NL;NL;
    for(p=&a[0],i=1;i<=5;i++){
        PR(d,*(p+i));
    }
    NL;
    for(p=a,i=0;p+i<=a+4;p++,i++){
        PR(d,*(p+i));
    }
    NL;NL;
    for(p=a+4;p>=a;p-- ) PR(d,*p);
    NL;
    for(p=a+4,i=0;i<=4;i++){PR(d,p[-i]);}
    NL;
    for(p=a+4;p>=a;p-- ){PR(d,a[p-a]);}
    NL;
}
```

Pointers and Arrays 2 : Array of Pointers

What does the following program print?

```
#include "defs.h"
int a[]={0,1,2,3,4};
int *p[]={a,a+1,a+2,a+3,a+4};
int **pp=p;

void main()
{
    PRINT2(d,a,*a);
    PRINT3(d,p,*p,**p);
    PRINT3(d,pp,*pp,**pp);
    NL;

    pp++;PRINT3(d,pp-p,*pp-a,**pp);
    *pp++;PRINT3(d,pp-p,*pp-a,**pp);
    **++pp;PRINT3(d,pp-p,*pp-a,**pp);
    ++*pp;PRINT3(d,pp-p,*pp-a,**pp);
    NL;

    pp=p;
    **pp++; PRINT3(d,pp-p,*pp-a,**pp);
    **++pp; PRINT3(d,pp-p,*pp-a,**pp);
    ++**pp; PRINT3(d,pp-p,*pp-a,**pp);
}
```


Pointer and Arrays 3 : Multidimensional Array

What does the following program print?

```
#include "defs.h"
int a[3][3]={ {1,2,3},{4,5,6},{7,8,9}};
int *pa[3]={a[0],a[1],a[2]};
int *p= a[0];

void main()
{
    int i;
    for(i=0;i<3;i++)
        PRINT3(d,a[i][2-i],*a[i],*(*(a+i)+i));
    NL;
    for(i=0;i<3;i++)
        PRINT2(d,*pa[i],p[i]);
}
```

Pointers and Arrays 4 : Pointer Stew

What does the following program print?

```
#include "defs.h"
char *c[]={"ENTER","NEW","POINT","FIRST"};
char **cp[]={c+3,c+2,c+1,c};
char ***cpp= cp;
void main()
{
    printf("%s", *** + cpp);
    printf("%s",*- - ** + cpp+ 3);
    printf("%s",*cpp[- 2]+ 3);
    printf("%s",cpp[- 1][- 1]+ 1);
}
```

Structures 1 : Simple Structures, Nested Structure

What does the following program print?

```
#include "defs.h"

void main()
{
    static struct S1{
        char c[4],*s;
    }s1={"abc","def"};
    static struct S2{
        char *cp;
        struct S1 ss1;
    } s2={"ghi",{ "jkl","mno"}};
    PRINT 2(c,s1.c[0],*s1.s);
    PRINT 2(s,s1.c,s1.s);
    PRINT 2(s,s2.cp,s2.ss1.s);
    PRINT 2(s,++s2.cp,++s2.ss1.s);
}
```

Structures 2 : Array of Structures

What does the following program print?

```
#include "defs.h"

struct S1{
    char *s;
    int i;
    struct S1 *s1p;
};

void main()
{
    static struct S1 a[] = {{ "abcd",1,a+1},{ "efgh",2,a+2},{ "ijkl",3,a}};
    struct S1 *p = a;
    int i;
    PRINT 3(s,a[0].s,p->s,a[2].s1p->s);
    for(i=0;i<2;i++){
        PR(d,--a[i].i);
        NL;
    }
    PRINT 3(s,++(p->s),a[(++p)->i].s,a[--(p->s1p->i)].s);
}
```

Structures 3 : Array of Pointers to Structures

What does the following program print?

```
#include "defs.h"
struct S1{
    char *s;
    struct S1 *s1p;
};

void swap(struct S1 *p1,struct S1 *p2);

void main()
{
    static struct S1 a[]={{"abcd",a+1},{ "efgh",a+2},{ "ijkl",a}};
    struct S1 *p[3];
    int i;
    for(i=0;i<3; i++){
        p[i]=a[i].s1p;
    }
    PRINT3(s,p[0]->s,(*p)->s,(**p).s);
    swap(*p,a);
    PRINT3(s,p[0]->s,(*p)->s, (*p)->s1p->s);
    swap(p[0],p[0]->s1p);
    PRINT3(s,p[0]->s,(*++p[0]).s,++(**+(*p)->s1p).s);
}

void swap(struct S1 *p1,struct S1 *p2)
{
    struct S1 temp;
    temp.s=p1->s;
    p1->s=p2->s;
    p2->s=temp.s;
}
```

Preprocessor 1 : The Preprocessor Knows Little C

What does the following program print?

```
#include <stdio.h>
#define FUDGE(k) k+ 3.14159
#define PR(a) printf("#a"= %dW',(int)(a))
#define PRINT(a) PR(a);putchar('W')
#define PRINT2(a,b) PR(a);PRINT(b)
#define PRINT3(a,b,c) PR(a);PRINT2(b,c)
#define MAX(a,b) (a<b ? b : a)

void main()
{
    {
        int x=2;
        PRINT(x*FUDGE(2));
    }
    {
        int cel;
        for(cel=0;cel<=100;cel+=50){
            PRINT2(cel,9./5*cel+32);
        }
    }
    {
        int x=1,y=2;
        PRINT3(MAX(x++,y),x,y);
        PRINT3(MAX(x++,y),x,y);
    }
}
```

preprocessor 2 : Caution Pays

What does the following program print?

```
#include <stdio.h>
#define weeks(mins) (days(mins)/7)
#define days(mins) (hours(mins)/24)
#define hours(mins) (mins/60)
#define mins(secs) (secs/60)
#define PRINT(a) printf("#a"= %d\n", (int)(a))
#define TRACE(x) if(traceon) printf("Trace: "),PRINT(x)
#define g(a,b) a a ## b(nd)
#define oo "th"
#define oodbye(a) "e e"# a

int traceon;

void main()
{
    {
        PRINT(weeks(10080));
        PRINT(days(mins(86400)));
    }
    {
        int i;
        traceon = 1;
        for(i=20; i>0; i/=2){
            if(i<10){
                TRACE(i);
            }
            else{
                puts("not yet");
            }
        }
    }
}
```

```
}
{
    puts(g(oo,dbye));
}
}
```