Chapter VII.

Experience is the best teacher.

, debugging,   .

(sub-program)

. main{}   (C

main{}   )   printf(), scanf(),

getchar()   (library function)   .

(programmer)   (user define function)

.

7.1   ?

(function)   .

(1)

(call)

2)   (modularization)   .

.

```
#include <stdio.h>
main()
{
    int a, b, c, d;
    read_data(&a,&b);
    c=add(a,b);
    d=average(a,b);
}
```

main()   (read_data(), add(), average())

.   (   )

.   read_data(), add(), average()   .

Black Box   .

.

┌───┐
│   math.c   .   math.c   │
│ include   .   │
└───┘

||EXAMPLE|| (1) $\sqrt{3}$? $x = sqrt(3)$ (2) 4^3 ? $x = pow(4,3)$ (3) e^5 ? $x = \exp(5)$

```c
#include <stdio.h>
#include <math.h>
main()
{
float x, y;
printf("지수 계산 (x^y): ");
scanf("%f %f",&x,&y);
printf("%5.2f^%5.2f=%10.4f\n",x,y,pow(x,y));
}
```

```
지수 계산 (x^y): 2.1 5.2
 2.10^ 5.20=   47.3740
Press any key to continue
```

```
abs   (int x);
acos  (double x);
asin  (double x);
atan  (double x);
atan2 (double y, double x);
atof  (const char *s);
ceil  (double x);
cos   (double x);
cosh  (double x);
exp   (double x);
fabs  (double x);
floor (double x);
fmod  (double x, double y);
frexp (double x, int *exponent);
labs  (long x);
ldexp (double x, int exponent);
log   (double x);
log10 (double x);
modf  (double x, double *ipart);
pow   (double x, double y);
sin   (double x);
sinh  (double x);
sqrt  (double x);
tan   (double x);
tanh  (double x);
```

7.2 　　　　(definition)

```
                              (              )
{
              ;

}
```

```
int add (int x, int y)
{
   int sum;
   sum=x+y;
   reteun sum;
}
```

(int), (float), (char) . int .

return . (cf) (call)

c=add(a,b) .

.

(argument)

. c=add(a,b) a x , b y add {} .

.

.

.

.

7.3

.

(calling function),

(called function)   .

return   .

C   main   main
        . main()

  .   .

  main()   for   line1()   10   line1()
                (printf()   )   .
                        ()   .

```c
#include <stdio.h>
#include <math.h>
main()
{int i;
for(i=1;i<=10;i=i+1){
line1();
printf("%d^2=%7.0f\n",i,pow(i,2));
}
}
line1()
{
printf("------------------\n");
}
```

```
------------------
1^2=        1
------------------
2^2=        4
------------------
3^2=        9
------------------
4^2=       16
------------------
5^2=       25
------------------
6^2=       36
```

EXERCISE 7-1   : 1024_FOR3.C

---

2   100   for()   .

```
정수  제곱근    제곱
 2   0.50       4
 4   0.25      16
 6   0.17      36
 8   0.13      64
10   0.10     100
12   0.08     144
14   0.07     196
```

---

main()   max()   a   x   b   m   .
                add   mx   m   m
                 . max()   main()   (mx)
            .

```c
#include <stdio.h>
main()
{int i,n,x, m=-999;
printf("데이터 개수=");
scanf("%d",&n);
for(i=1;i<=n;i=i+1){
printf("%d번째 데이터=",i);
scanf("%d",&x);
m=max(x,m);
}
printf("최대값=%-7d\n",m);
}
int max(int a, int b)
{int mx;
mx=(a>=b)?a:b;
return mx;
}
```

```
데이터 개수=4
1번째 데이터=1
2번째 데이터=2
3번째 데이터=5
4번째 데이터=3
최대값=5
Press any key t
```

EXERCISE 7-2   : 1024_NF.C

Factorial(!)   (FAC)   n!   .

```c
#include <stdio.h>
main()
{int n, nu;
printf("n!? ");
scanf("%d",&n);
nu=fac(n);
printf("%d!=%d\n",n,nu);
```

```
n!? 5
5!=120
Press any key
```

EXERCISE 7-3   : 1024_CMN.C

Factorial(!)   (FAC) $_nC_r = \dfrac{n!}{(n-r)!r!}$

            .

```
n과 r을 입력하시오:6 3
6C3=20
Press any key to contin
```

$(a+b)^n$

<math.h>   pow()   . main()

a, b, n   sq1   x, y, z   . sq1   sq2

. sq2   s, z   x, y   .   x, y   sq1   z, y

. sq2   mul   sq1   res   sq1   res   main

c   .

```
#include <stdio.h>
#include <math.h>
main()
{
    int a,b, c,res,n;
    printf("Two number:");
    scanf("%d %d",&a,&b);
    printf("n?");
    scanf("%d",&n);
    c=sq1(a,b,n);
    printf("(%d+%d)^%d=%d",a,b,n,c);
}
```

```
int sq1(int x, int y, int z)
{
    int s,res;
    s=x+y;
    res=sq2(s,z);
    return res;
}
int sq2(int x, int y)
{
    int mul;
    mul=pow(x,y);
    return mul;
}
```

main() 함수	sq1() 함수	sq2() 함수
sq1() 함수 호출 a, b, n	sq2() 함수 호출 s, z	mul 값 sq1 함수 내의 변수 res 로 보냄
c 라인 다음 printf()문 실행	res 값 main 함수 내의 변수 c 로 보냄	

EXERCISE 7-1   : 1024_FOR3.C

2   100   for()   .

```c
#include <stdio.h>
#include <math.h>
main()
{int i;
printf("정수  제곱근   제곱\n");
for(i=2;i<=100;i=i+2){
    printf("%3d %5.2f %6d\n", i,1.0/i, i*i);
}
}
```

```
정수  제곱근   제곱
  2  0.50      4
  4  0.25     16
  6  0.17     36
  8  0.13     64
 10  0.10    100
 12  0.08    144
 14  0.07    196
```

EXERCISE 7-1   : 1024_CMN.C

Factorial(!)   (FAC)   n!   .

```c
#include <stdio.h>
main()
{int n, nu;
printf("n!? ");
scanf("%d",&n);
nu=fac(n);
printf("%d!=%d\n",n,nu);
}
int fac(int a)
{int m=1,i;
for(i=1;i<=a;i=i+1){m=m*i;}
return m;
}
```

```
n!? 5
5!=120
Press any key
```

EXERCISE 7-2   : 1024_CMN.C

Factorial(!)   (FAC) $_nC_r = \dfrac{n!}{(n-r)!r!}$

.

```c
#include <stdio.h>
main()
{int n, r,nu,dn1,dn2;
printf("n과 r을 입력하시오:");
scanf("%d %d",&n,&r);
nu=fac(n);
dn1=fac(n-r);
dn2=fac(r);
printf("%dC%d=%d\n",n,r,nu/dn1/dn2);
}
int fac(int a)
{int m=1,i;
for(i=1;i<=a;i=i+1){m=m*i;}
return m;
}
```
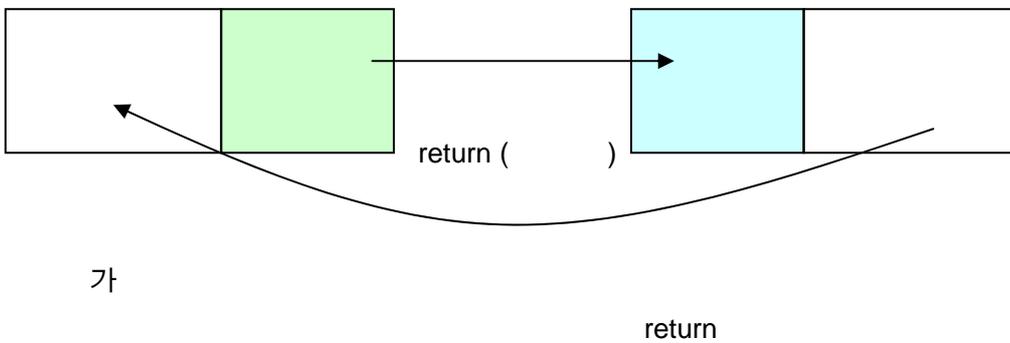
```
n과 r을 입력하시오:6 3
6C3=20
Press any key to contir
```

7.4

Debugging

.

.

.   (calling function)   (called
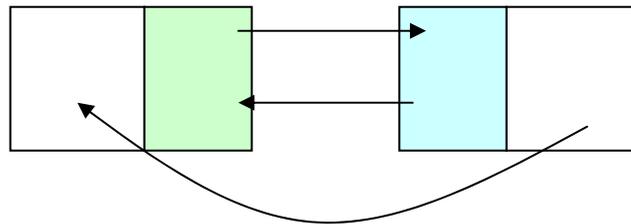function)   (call by value)
(pointer:)   .   .

7.4.1

return (   )

return

.

.

```
#include <stdio.h>
main()
{
    int i=10, j=0;
    j=inc(i);
    printf("\n %d %d",i,j);
}
int inc(int k)
{
    k+=k;
    printf("\n %d",k);
    return k;
}
```

```
#include <stdio.h>
main()
{
    int i=10, j=0;
    inc(i,j);
    printf("\n %d %d",i,j);
}
int inc(int i,int j)
{
    i=3;j=5;
}
```

```
#include <stdio.h>
main()
{
    int i=10, j=0;
    inc(i,j);
    printf("\n %d %d",i,j);
}
int inc(int i,int j)
{
    i=i+(j+3);
    return i;
}
```

```
#include <stdio.h>
main()
{
    int i=10, j=0, k=5;
    k=inc(i,j+k);
    printf("\n %d %d %d",i,j,k);
}
int inc(int a, int b)
{
    return(a*b);
}
```

7.4.2

.

(address: pointer)

.

pointer   .

```
#include <stdio.h>
main()
{
    int x=1, y=2;
    y=sw(x,y);
    printf("x=%d, y=%d",x,y);
}
sw(int x, int y)
{
    return x;
}
```

(   )

```
#include <stdio.h>
main()
{
    int x=1, y=2;
    sw(x,y);
    printf("x=%d, y=%d",x,y);
}
sw(int x, int y)
{
    int temp; temp=x; x=y; y=temp;
}
```

(   )

```
#include <stdio.h>
main()
{
    int x=1, y=2;
    sw(&x,&y);
    printf("x=%d, y=%d",x,y);
}
sw(int *x, int *y)
{
    int temp; temp=*x; *x=*y; *y=temp;
}
```

main()	&x &y	x y
sw()	*x *y	*x *y

1:                    ➔

.

local   .

main()	s=(x, y)	s m
		x, y .
sw()	(x , y) ➔ m	return m

2:                    ➔

(pointer)

.

        &   *   . &

  global   .

main()	&x &y	x y ( )
sw()	*x *y	*x *y

```
#include <stdio.h>
main()
{
   int a=10,b=20, c=0;
   c=add(a,b);
   printf("%d+%d=%d",a,b,c);
}
int add(???)
{
???????
}
```

```
#include <stdio.h>
main()
{
   int a=10,b=20, c=0;
   add(???);
   printf("%d+%d=%d",a,b,???);
}
int add(???)
{
???????
}
```

```
#define k 1
#include <stdio.h>
main()
{
   int a=1,b=2, c=4, d;
   d=abc(&a,b,c);
   printf("a=%d, b=%d, c=%d, d=%d",a,b,c,d);
}
int abc(int *x, int y, int z)
{
*x=y+z;
y=y+z;
return(k);
}
```