

```
/* distance1.c */

#include <stdio.h>
#include <math.h>

struct point{
    int x;
    int y;
};

int main(void)
{
    struct point p1, p2;
    double distance;

    fputs("첫번째 점의 x, y 좌표 입력 :", stdout);
    scanf("%d %d", &p1.x, &p1.y);

    fputs("두번째 점의 x, y 좌표 입력 :", stdout);
    scanf("%d %d", &p2.x, &p2.y);

    /* 두 점간의 거리 계산 공식 */
    distance =
        sqrt((p1.x-p2.x)*(p1.x-p2.x)+(p1.y-p2.y)*(p1.y-p2.y));

    printf("두 점의 거리는 %f 입니다", distance);

    return 0;
}
```

```
/* TelPhone2.c */
```

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

```
struct person {  
    char name[20];  
    char phone[20];  
};
```

```
int main (void)
```

```
{  
    struct person pArray[3];  
    int i;  
  
    for(i=0; i<3; i++)    //데이터 입력.  
    {  
        printf("이름, 전화번호 순으로 입력 :");  
        scanf("%s %s", pArray[i].name, pArray[i].phone);  
    }  
  
    printf("\n입력 결과는 다음과 같습니다.\n");  
    for(i=0; i<3; i++)    //데이터 출력.  
    {  
        printf("이름 : %s, ", pArray[i].name);  
        printf("전화번호 : %s\n", pArray[i].phone);  
    }  
  
    return 0;  
}
```

```
/* struct_pointer1.c */
```

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

```
struct person {  
    char name[20];  
    char phone[20];  
};
```

```
int main()  
{
```

```
    struct person man={"Thomas", "354-00xx"};  
    struct person * pMan;  
    pMan=&man;
```

```
    // 구조체 변수를 이용한 출력.
```

```
    printf("name : %s\n", man.name);  
    printf("phone : %s\n", man.phone);
```

```
    // 구조체 포인터를 이용한 출력1.
```

```
    printf("name : %s\n", (*pMan).name);  
    printf("phone : %s\n", (*pMan).phone);
```

```
    // 구조체 포인터를 이용한 출력2.
```

```
    printf("name : %s\n", pMan->name);  
    printf("phone : %s\n", pMan->phone);
```

```
    return 0;
```

```
}
```

```
/* struct_pointer2.c */

#include <stdio.h>

struct perInfo {
    char addr[30];
    char tel[20];
};

struct person {
    char name[20];
    char pID[20];
    struct perInfo* info;
};

int main()
{
    struct perInfo info={"Korea Seoul", "333-4444"};
    struct person man={"Mr. Lee", "820204-xxxx512"};

    man.info=&info;

    printf("name : %s\n", man.name);
    printf("pID  : %s\n", man.pID);
    printf("addr : %s\n", man.info->addr);
    printf("tel  : %s\n", man.info->tel);

    return 0;
}
```

```
/* struct_pointer3.c */
```

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

```
struct person {  
    char name[20];  
    char pID[20];  
    struct person* frnd;  
};
```

```
int main()
```

```
{  
    struct person man1={"Mr. Lee", "820204-0000512"};  
    struct person man2={"Mr. Lee's Friend", "820000-0000101"};  
  
    man1.frnd=&man2;  
  
    printf("[Mr. Lee]\n");  
    printf("name : %s\n", man1.name);  
    printf("pID   : %s\n", man1.pID);  
  
    printf("[His Friend]\n");  
    printf("name : %s\n", man1.frnd->name);  
    printf("pID   : %s\n", man1.frnd->pID);  
  
    return 0;  
}
```

```

/* struct_callby.c */
#include <stdio.h>

struct simple {
    int data1;
    int data2;
};

void show(struct simple ts); // call-by-value
void swap(struct simple * ps); // call-by-reference

int main()
{
    struct simple s={1, 2};

    show(s); // s의 멤버 출력
    swap(&s); // s의 멤버 data1, data2의 값 변경
    show(s); // s의 변경된 멤버 출력

    return 0;
}

void show(struct simple ts) // call-by-value
{
    printf("data1:%d, data2:%d\n", ts.data1, ts.data2);
}

void swap(struct simple * ps) // call-by-reference
{
    int temp;
    temp=ps->data1;
    ps->data1=ps->data2;
    ps->data2=temp;
}

```

```
/* overlapped.c */
#include <stdio.h>

struct point{
    int x;
    int y;
};

struct circle {
    struct point p;
    double radius;
};

int main()
{
    struct circle c1={10, 10, 1.5};
    struct circle c2={{30, 30}, 2.4};

    printf("[circle1] \n");
    printf("x:%d, y:%d \n", c1.p.x, c1.p.y);
    printf("radius:%f \n",c1.radius);

    printf("[circle2] \n");
    printf("x:%d, y:%d \n", c2.p.x, c2.p.y);
    printf("radius:%f \n",c2.radius);

    return 0;
}
```

```

/* typedef.c */
#include <stdio.h>

typedef int INT;
typedef int * P_INT;

typedef unsigned int UINT;
typedef unsigned int * P_UINT;

typedef unsigned char UCHAR;
typedef unsigned char * P_UCHAR;

struct Data {
    int data1;
    int data2;
};

typedef struct Data Data;

int main(void)
{
    INT a=10;          //int a=10;
    P_INT pA=&a;      //int * pA=&a;

    UINT b=100;       //unsigned int b=100;
    P_UINT pB=&b;     //unsigned int * pB=&b;

    UCHAR c='a';      //unsigned char c='a';
    P_UCHAR pC=&c;    //unsigned char * pC=&c;

    Data d = {1, 2};

    printf("%d, %d, %c \n", *pA, *pB, *pC);
    printf("%d, %d\n", d.data1, d.data2);

    return 0;
}

```



```
/* union.c */
#include <stdio.h>

union u_data{
    int d1;
    double d2;
    char d3;
};

int main (void)
{
    union u_data data;

    data.d2=3.3;
    printf("%d, %f, %c \n", data.d1, data.d2, data.d3);

    data.d1=2;
    printf("%d, %f, %c \n", data.d1, data.d2, data.d3);

    data.d3='a';
    printf("%d, %f, %c \n", data.d1, data.d2, data.d3);

    return 0;
}
```

```
/* enum_usable1.c */
#include <stdio.h>

int main (void)
{
    int day;
    printf("Input a day(0:Monday ~ 6:Sunday) : ");
    scanf("%d", &day);

    switch(day) {
    case 0 :
        printf("수영을 하세요. \n");
        break;
    case 1 :
        printf("조깅을 하세요. \n");
        break;
    case 2 :
        printf("조금 더 주무세요. \n");
        break;
    case 3 :
        printf("볼링을 하세요. \n");
        break;
    case 4 :
        printf("조금 더 주무세요. \n");
        break;
    case 5 :
        printf("등산을 하세요. \n");
        break;
    case 6 :
        printf("원하는걸 하세요. \n");
        break;
    }

    return 0;
}
```

```
/* enum_usable2.c*/
#include <stdio.h>

enum days {MON, TUE, WED, THU, FRI, SAT, SUN};

int main (void)
{
    enum days day;
    printf("Input a day(0:Monday ~ 6:Sunday) : ");
    scanf("%d", &day);

    switch(day)
    {
    case MON :
        printf("수영을 하세요. \n");
        break;
    case TUE :
        printf("조깅을 하세요. \n");
        break;
    case WED :
        printf("조금 더 주무세요. \n");
        break;
    case THU :
        printf("볼링을 하세요. \n");
        break;
    case FRI :
        printf("조금 더 주무세요. \n");
        break;
    case SAT :
        printf("등산을 하세요. \n");
        break;
    case SUN :
        printf("원하는걸 하세요. \n");
        break;
    }
}
```

```
} return 0;
```