```
/* Demonstrates using arrays of structures. */

#include <stdio.h>

/* Define a structure to hold entries. */

struct entry {
    char fname[20];
    char lname[20];
    char phone[10];
};

/* Declare an array of structures. */

struct entry list[4];

int i;

int main(void)
{
    /* Loop to input data for four people. */

    for (i = 0; i < 4; i++)
    {
        printf("\nEnter first name: ");
        scanf("%s", list[i].fname);
        printf("Enter last name: ");
        scanf("%s", list[i].lname);
        printf("Enter phone in 123-4567 format: ");
        scanf("%s", list[i].phone);
    }

    /* Print two blank lines. */

    printf("\n\n");

    /* Loop to display data. */

    for (i = 0; i < 4; i++)
    {
        printf("Name: %s %s", list[i].fname, list[i].lname);
        printf("\t\tPhone: %s\n", list[i].phone);
    }
}
```

```
/* Demonstrates stepping through an array of structures */
/* using pointer notation. */

#include <stdio.h>

#define MAX 4

/* Define a structure, then declare and initialize */
/* an array of 4 structures. */

struct part {
    int number;
    char name[10];
};
/* Declare a pointer to type part, and a counter variable. */

struct part data[MAX] =
                {
                    {1, "Smith"},
                    {2, "Jones"},
                    {3, "Adams"},
                    {4, "Wilson"}
                };

/* Declare a pointer to type part, and a counter variable. */


struct part *p_part;
int count;

int main(void)
{
    /* Initialize the pointer to the first array element. */

    p_part = data;

    /* Loop through the array, incrementing the pointer */
    /* with each iteration. */

    for (count = 0; count < MAX; count++)
    {
        printf("\nAt address %p: %d %s", p_part, p_part->number,
                p_part->name);
        p_part++;
    }

}
```

```
/* Demonstrates structures that contain other structures.
 */

/* Receives input for corner coordinates of a rectangle and
   calculates the area. Assumes that the y coordinate of the
   upper-left corner is greater than the y coordinate of the
   lower-right corner, that the x coordinate of the lower-
   right corner is greater than the x coordinate of the upper-
   left corner, and that all coordinates are positive. */

#include <stdio.h>

int length, width;
long area;

struct coord{
    int x;
    int y;
};

struct rectangle{
    struct coord *topleft;
    struct coord *bottomrt;
};

int main(void)
{
    // create the object
      rectangle *mybox = new rectangle;

      mybox->topleft = new coord;
      mybox->bottomrt = new coord;

    /* Input the coordinates */

    printf("\nEnter the top left x coordinate: ");
    scanf("%d", &mybox->topleft->x);

    printf("\nEnter the top left y coordinate: ");
    scanf("%d", &mybox->topleft->y);

    printf("\nEnter the bottom right x coordinate: ");
    scanf("%d", &mybox->bottomrt->x);

    printf("\nEnter the bottom right y coordinate: ");
    scanf("%d", &mybox->bottomrt->y);

    /* Calculate the length and width */

    width = mybox->bottomrt->x - mybox->topleft->x;
    length = mybox->bottomrt->y - mybox->topleft->y;

    /* Calculate and display the area */

    area = width * length;
    printf("The area is %ld units.", area);

    return 0;
}
```