```c
// Write a C program to swap two numbers. Make two
functions named as call_by_value
// and call_by_reference to pass the
value/reference of two variables for swapping.
#include <stdio.h>

void swap_by_value(int a, int b) {
    int temp = a; a = b; b = temp;
    printf("Inside swap_by_value: a = %d, b =
%d\n", a, b);
}

void swap_by_reference(int *a, int *b) {
    int temp = *a; *a = *b; *b = temp;
}

int main() {
    int x, y;
    printf("Enter x and y: ");
    scanf("%d %d", &x, &y);

    printf("Before swap_by_value: x = %d, y =
%d\n", x, y);
    swap_by_value(x, y);
    printf("After swap_by_value: x = %d, y = %d\n",
x, y);

    swap_by_reference(&x, &y);
```

```c
    printf("After swap_by_reference: x = %d, y =
%d\n", x, y);
    return 0;
}


// 2. Write a C program to create an array having
10 elements and initialize it with numbers 1
// to 10. Print the array. Take a pointer, say p,
point to the base address, and loop through
// the addresses to access each element address and
increase the value at the address of
// each element by 2. Again, print the elements of
the array.
#include <stdio.h>

int main() {
    int arr[10], *p = arr;
    for (int i = 0; i < 10; i++) *(p + i) = i + 1;

    printf("Original: ");
    for (int i = 0; i < 10; i++) printf("%d ", *(p
+ i));

    for (int i = 0; i < 10; i++) *(p + i) += 2;

    printf("\nModified: ");
```

```c
    for (int i = 0; i < 10; i++) printf("%d ", *(p
+ i));
    return 0;
}

// 3. Take a 3-dimensional array, say a[p][r][c],
where p denotes planes, each with r rows and
// c columns. Initialize the array with numbers
sequentially from 1 to p*r*c, and take a
// pointer, x, pointing to the base address of the
array. Write a C program to print the last
// element of each plane accessed through the
pointer.
#include <stdio.h>

int main() {
    int p = 3, r = 3, c = 3;
    int a[p][r][c];
    int element = 1;

    // Fill array with sequential values
    for (int i = 0; i < p; i++)
        for (int j = 0; j < r; j++)
            for (int k = 0; k < c; k++)
                a[i][j][k] = element++;

    int *x = &a[0][0][0];
```

```c
    // Print last element of each plane
    for (int i = 0; i < p; i++) {
        int index = (i * r * c) + (r * c - 1);
        printf("Last element of plane %d: %d\n", i,
*(x + index));
    }

    return 0;
}


// 4. Write a C program to create an array with n
elements. For the user-given input, x, rotate
// the array elements to the left with x positions.
#include <stdio.h>

int main() {
    int n, x;
    printf("Enter number of elements in the array:
");
    scanf("%d", &n);

    int a[n];
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);
```

```c
    printf("Enter number of positions to rotate left: ");
    scanf("%d", &x);

    x %= n; // Normalize x

    printf("Rotated array:\n");
    for (int i = 0; i < n; i++)
        printf("%d ", a[(i + x) % n]);

    return 0;
}


// 5. Write a C program to create a grade sheet using structures containing names, roll
// numbers, marks, and grades of 5 students. The program should be able to insert a new
// record, delete it, and modify it based on user input, i.e., menu-driven.
#include <stdio.h>
#include <string.h>

#define MAX 100

typedef struct {
    char name[30];
    int roll;
```

```c
    float marks;
    char grade;
} Student;

Student s[MAX];
int count;

void input(int i) {
    printf("Name Roll Marks Grade: ");
    scanf("%s %d %f %c", s[i].name, &s[i].roll,
&s[i].marks, &s[i].grade);
}

void display() {
    for (int i = 0; i < count; i++)
        printf("%d. %s | %d | %.1f | %c\n", i + 1,
s[i].name, s[i].roll, s[i].marks, s[i].grade);
}

int main() {
    printf("How many student records do you want to
enter? ");
    scanf("%d", &count);

    for (int i = 0; i < count; i++) input(i);

    while (1) {
        int ch, pos;
```

```
        printf("\n1.Insert 2.Delete 3.Modify 4.Show
5.Exit: ");
        scanf("%d", &ch);

        if (ch == 1 && count < MAX) input(count++);
        else if (ch == 2) {
            printf("Delete index (0 to %d): ",
count - 1);
            scanf("%d", &pos);
            if (pos >= 0 && pos < count) {
                for (int i = pos; i < count - 1;
i++) s[i] = s[i + 1];
                count--;
            }
        }
        else if (ch == 3) {
            printf("Modify index (0 to %d): ",
count - 1);
            scanf("%d", &pos);
            if (pos >= 0 && pos < count)
input(pos);
        }
        else if (ch == 4) display();
        else if (ch == 5) break;
    }
    return 0;
}
```