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

struct Node
{
    int data;
    struct Node *next;
};

struct Node *head = NULL, *tail = NULL;

void addNode(int data)
{
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    if (head == NULL)
    {
        head = newNode;
        tail = newNode;
    }
    else
    {
        tail->next = newNode;
        tail = newNode;
    }
}

void display()
{
    struct Node *current = head;
    if (head == NULL)
    {
        printf("List is empty.\n");
    }
    else
    {
        printf("The Nodes of the list are: ");
        while (current != NULL)
        {
            printf("%d ", current->data);
            current = current->next;
        }
        printf("\n");
    }
}

void delete_begin()
{
    if (head == NULL)
    {
        printf("The list is empty.");
    }
    else
    {
        struct Node *temp = head;
        head = head->next;
        free(temp);
    }
}

void delete_end()
{
    if (head == NULL)
    {
        printf("List is empty.");
    }
    else
    {
        struct Node *end = head;
        struct Node *prev = NULL;
        while (end->next)
        {
            prev = end;
            end = end->next;
        }
        prev->next = NULL;
        tail = prev;
        free(end);
    }
}

void delete_at(int position)
{
    if (head == NULL)
    {
        printf("List is empty.");
    }
    else
    {
        struct Node *temp;
        struct Node *prev;
        temp = head;
        prev = head;
        for (int i = 0; i < position; i++)
        {
            if (i == 0 && position == 1)
            {
                head = head->next;
                free(temp);
            }
            else
            {
                if (i == position - 1 && temp)
                {
                    prev->next = temp->next;
                    free(temp);
                }
                else
                {
                    prev = temp;
                    if (prev == NULL)
                    {
                        printf("Position does not exist.");
                        break;
                    }
                    temp = temp->next;
                }
            }
        }
    }
}

void delete_val(int value)
{
    if(head == NULL)
    {
        printf("List is empty.");
    }
    else
    {
        struct Node *temp;
        temp = head;
        if(temp->data == value)
        {
            delete_begin();
        }
        else
        {
            while(temp != NULL)
            {
                if(temp->next->data == value)
                {
                    temp->next = temp->next->next;
                    free(temp);
                    break;
                }
                else
                {
                    temp = temp->next;
                }
            }
        }
    }
}

int main()
{
    int ch = 0, n, f = 0, ff = 0;
    while (f == 0)
    {
        if(ff == 0)
        {
            printf("Enter 1 to delete element at top, 2 to delete element at end,
3 to delete element at position, 4 to delete a value, 5 to add element, 6 to
display elements, 7 to exit the code: ");
            ff = 1;
        }
        else
        {
            printf("Enter your choice: ");
        }
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            delete_begin();
            break;
        case 2:
            delete_end();
            break;
        case 3:
            printf("Enter the position to delete: ");
            scanf("%d", &n);
            delete_at(n);
            break;
        case 4:
            printf("Enter the value to delete: ");
            scanf("%d", &n);
            delete_val(n);
            break;
        case 5:
            printf("Enter a number: ");
            scanf("%d", &n);
            addNode(n);
            break;
        case 6:
            display();
            break;
        case 7:
            printf("Exiting code execution...");
            f = 1;
            break;
        default:
            printf("Wrong choice!!");
        }
    }
    return 0;
}
```