

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node {
5      int data;
6      struct node *next;
7  };
8
9  struct node *head = NULL;
10
11
12 void insert_end(int data) {
13     struct node *newNode = (struct node*)malloc(sizeof(struct node));
14     newNode->data = data;
15     newNode->next = NULL;
16
17     if (head == NULL) {
18         head = newNode;
19         return;
20     }
21
22     struct node *temp = head;
23     while (temp->next != NULL)
24         temp = temp->next;
25
26     temp->next = newNode;
27 }
28
29
30 void delete_beg() {
31     if (head == NULL) {
32         printf("\nList is empty.\n");
33     } else {
34         struct node *temp = head;
35         head = head->next;
36         free(temp);
37         printf("\nNode deleted from beginning.\n");
38     }
39 }
40
41

```

```

42     free(temp);
43     printf("\nNode deleted from beginning.\n");
44 }
45
46
47 void delete_end() {
48     if (head == NULL) {
49         printf("\nList is empty.\n");
50     } else if (head->next == NULL) {
51         free(head);
52         head = NULL;
53         printf("\nOnly node deleted.\n");
54     } else {
55         struct node *temp = head;
56         struct node *prev = NULL;
57
58         while (temp->next != NULL) {
59             prev = temp;
60             temp = temp->next;
61         }
62
63         prev->next = NULL;
64         free(temp);
65         printf("\nNode deleted from end.\n");
66     }
67 }
68
69
70 void delete_pos() {
71     if (head == NULL) {
72         printf("\nList is empty.\n");
73         return;
74     }
75 }
76

```

```

void delete_pos() {
    if (head == NULL) {
        printf("\nList is empty.\n");
        return;
    }

    int pos;
    printf("\nEnter position to delete: ");
    scanf("%d", &pos);

    if (pos <= 0) {
        printf("\nInvalid position.\n");
        return;
    }

    if (pos == 1) {
        struct node *temp = head;
        head = head->next;
        free(temp);
        printf("\nNode deleted from position 1.\n");
        return;
    }

    struct node *temp = head;
    struct node *prev = NULL;

    for (int i = 1; i < pos && temp != NULL; i++) {
        prev = temp;
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("\nPosition out of range.\n");
        return;
    }

    prev->next = temp->next;
    free(temp);
    printf("\nNode deleted from position %d.\n", pos);
}

```

```
void display() {  
    struct node *temp = head;  
    if (temp == NULL) {  
        printf("\nList is empty.\n");  
        return;  
    }  
  
    printf("\nLinked List: ");  
    while (temp != NULL) {  
        printf("%d -> ", temp->data);  
        temp = temp->next;  
    }  
    printf("NULL\n");  
}
```

```

120
121
122 int main() {
123     int choice, data;
124
125     while (1) {
126         printf("\n\n--- Linked List Menu ---");
127         printf("\n1. Insert at End");
128         printf("\n2. Display");
129         printf("\n3. Delete from Beginning");
130         printf("\n4. Delete from End");
131         printf("\n5. Delete from Specific Position");
132         printf("\n6. Exit");
133
134         printf("\nEnter your choice: ");
135         scanf("%d", &choice);
136
137         switch (choice) {
138             case 1:
139                 printf("Enter data: ");
140                 scanf("%d", &data);
141                 insert_end(data);
142                 break;
143             case 2:
144                 display();
145                 break;
146             case 3:
147                 delete_beg();
148                 break;
149             case 4:
150                 delete_end();
151                 break;
152             case 5:
153                 delete_pos();
154                 break;
155             case 6:
156                 printf("\nExiting...\n");
157                 exit(0);
158             default:
159                 printf("\nInvalid choice!\n");
160         }
161     }
162
163     return 0;
164 }
165

```

```

--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 1
Enter data: 4

--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 16
Invalid choice!

--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 3
Node deleted from beginning.

--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 2
List is empty.

--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice:

```