

```

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

```

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

```