LAB PROGRAM 5

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

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

struct node *head = NULL;

void create() {
    int n, i, value;
    struct node *temp, *newnode;

    printf("Enter number of nodes: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        newnode = (struct node *) malloc (size of(struct node));
        printf("Enter data: ");
        scanf("%d", &value);
        newnode->data = value;
        newnode->next = NULL;

        if (head == NULL) {
            head = newnode;
```

```
            temp = head;
        } else {
            temp->next = newnode;
            temp = newnode;
        }
    }
}

void deleteFirst() {
    struct node *temp;
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    temp = head;
    head = head->next;
    free(temp);
}

void deleteLast() {
    struct node *temp, *prev;
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    if (head->next == NULL) {
        free(head);
        head = NULL;
        return;
```

```
    }
    temp = head;
    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    prev->next = NULL;
    free(temp);
}

void deleteSpecified() {
    int key;
    struct node *temp, *prev;

    if (head == NULL) {
        printf("List is empty\n");
        return;
    }

    printf("Enter element to delete: ");
    scanf("%d", &key);

    if (head->data == key) {
        temp = head;
        head = head->next;
        free(temp);
        return;
    }
```

```
    temp = head;
    while (temp != NULL && temp->data != key) {
        prev = temp;
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("Element not found\n");
    } else {
        prev->next = temp->next;
        free(temp);
    }
}

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

```
int main() {
    int choice;
    while (1) {
        printf("\n1.Create\n2.Delete First\n3.Delete Specified\n4.Delete
Last\n5.Display\n6.Exit\n");
        scanf("%d", &choice);

        switch (choice) {
            case 1: create(); break;
            case 2: deleteFirst(); break;
            case 3: deleteSpecified(); break;
            case 4: deleteLast(); break;
            case 5: display(); break;
            case 6: exit(0);
            default: printf("Invalid choice\n");
        }
    }
}
```

OUTPUT:

```
1.Create
2.Delete First
3.Delete Specified
4.Delete Last
5.Display
6.Exit
1

Enter number of nodes:3

Enter data:1

Enter data:2

Enter data:3
```

```
1.Create
2.Delete First
3.Delete Specified
4.Delete Last
5.Display
6.Exit
2


1.Create
2.Delete First
3.Delete Specified
4.Delete Last
5.Display
6.Exit
3

Enter element to delete:2
```

```
1.Create
2.Delete First
3.Delete Specified
4.Delete Last
5.Display
6.Exit
4


1.Create
2.Delete First
3.Delete Specified
4.Delete Last
5.Display
6.Exit
5

List is empty
```