**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* createNode(int data) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

newNode->data = data;

newNode->next = NULL;

return newNode;

}

void insertStart(struct Node\*\* head, int data) {

struct Node\* newNode = createNode(data);

newNode->next = \*head;

\*head = newNode;

}

void insertEnd(struct Node\*\* head, int data) {

struct Node\* newNode = createNode(data);

if (\*head == NULL) {

\*head = newNode;

return;

}

struct Node\* temp = \*head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

}

void insertBetween(struct Node\*\* head, int data, int X) {

struct Node\* newNode = createNode(data);

struct Node\* temp = \*head;

while (temp != NULL && temp->data != X) {

temp = temp->next;

}

if (temp != NULL) {

newNode->next = temp->next;

temp->next = newNode;

} else {

printf("Value %d not found in the list. Insertion failed.\n", X);

free(newNode);

}

}

void deleteStart(struct Node\*\* head) {

if (\*head == NULL) {

printf("List is empty. Deletion failed.\n");

return;

}

struct Node\* temp = \*head;

\*head = (\*head)->next;

free(temp);

}

void deleteEnd(struct Node\*\* head) {

if (\*head == NULL) {

printf("List is empty. Deletion failed.\n");

return;

}

struct Node\* temp = \*head;

struct Node\* prev = NULL;

if (temp->next == NULL) {

free(temp);

\*head = NULL;

return;

}

while (temp->next != NULL) {

prev = temp;

temp = temp->next;

}

free(temp);

prev->next = NULL;

}

void deleteAfter(struct Node\*\* head, int X) {

struct Node\* temp = \*head;

while (temp != NULL && temp->data != X) {

temp = temp->next;

}

if (temp != NULL && temp->next != NULL) {

struct Node\* toDelete = temp->next;

temp->next = toDelete->next;

free(toDelete);

} else {

printf("Value %d not found or does not have a next node. Deletion failed.\n", X);

}

}

void displayList(struct Node\* node) {

if (node == NULL) {

printf("List is empty.\n");

return;

}

while (node != NULL) {

printf("%d -> ", node->data);

node = node->next;

}

printf("NULL\n");

}

int main() {

struct Node\* head = NULL;

int choice, data, X;

while (1) {

printf("\nMenu:\n");

printf("1. Insert at Beginning\t\t");

printf("2. Insert at End\n");

printf("3. Insert Between Values\t");

printf("4. Delete from Beginning\n");

printf("5. Delete from End\t\t");

printf("6. Delete After Value\n");

printf("7. Display List\t\t\t");

printf("8. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter data to insert at beginning: ");

scanf("%d", &data);

insertStart(&head, data);

break;

case 2:

printf("Enter data to insert at end: ");

scanf("%d", &data);

insertEnd(&head, data);

break;

case 3:

printf("Enter data to insert: ");

scanf("%d", &data);

printf("Enter the value after which to insert: ");

scanf("%d", &X);

insertBetween(&head, data, X);

break;

case 4:

deleteStart(&head);

break;

case 5:

deleteEnd(&head);

break;

case 6:

printf("Enter the value after which to delete: ");

scanf("%d", &X);

deleteAfter(&head, X);

break;

case 7:

printf("Linked List: ");

displayList(head);

break;

case 8:

exit(0);

default:

printf("Invalid choice. Please try again.\n");

}

}

return 0;

}

**OUTPUT:**

PS D:\C Data Structure\9. Linked List> gcc Linkedlist.c

PS D:\C Data Structure\9. Linked List> ./a

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 1

Enter data to insert at beginning: 13

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 2

Enter data to insert at end: 23

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 3

Enter data to insert: 64

Enter the value after which to insert: 5

Value 5 not found in the list. Insertion failed.

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 7

Linked List: 13 -> 23 -> NULL

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 4

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 7

Linked List: 23 -> NULL

Menu:

1. Insert at Beginning 2. Insert at End

3. Insert Between Values 4. Delete from Beginning

5. Delete from End 6. Delete After Value

7. Display List 8. Exit

Enter your choice: 8

PS D:\C Data Structure\9. Linked List>