**10. Write a C program to implement Linked list operations**

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* head = NULL;

void insertAtBeginning(int data) {

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

newNode->data = data;

newNode->next = head;

head = newNode;

printf("Inserted %d at beginning\n", data);

}

void insertAtEnd(int data) {

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

newNode->data = data;

newNode->next = NULL;

if(head == NULL) {

head = newNode;

} else {

struct Node\* temp = head;

while(temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

}

printf("Inserted %d at end\n", data);

}

void deleteNode(int data) {

if(head == NULL) {

printf("List is empty\n");

return;

}

struct Node \*temp = head, \*prev = NULL;

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

head = temp->next;

free(temp);

printf("Deleted %d\n", data);

return;

}

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

prev = temp;

temp = temp->next;

}

if(temp == NULL) {

printf("%d not found in list\n", data);

return;

}

prev->next = temp->next;

free(temp);

printf("Deleted %d\n", data);

}

void search(int data) {

struct Node\* temp = head;

int position = 1;

while(temp != NULL) {

if(temp->data == data) {

printf("%d found at position %d\n", data, position);

return;

}

temp = temp->next;

position++;

}

printf("%d not found in list\n", data);

}

void display() {

if(head == NULL) {

printf("List is empty\n");

return;

}

struct Node\* temp = head;

printf("Linked List: ");

while(temp != NULL) {

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

if(temp->next != NULL) printf(" -> ");

temp = temp->next;

}

printf(" -> NULL\n");

}

int main() {

int choice, data;

while(1) {

printf("\nLinked List Operations:\n");

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

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

printf("3. Delete Node\n");

printf("4. Search Node\n");

printf("5. Display List\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

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

scanf("%d", &data);

insertAtBeginning(data);

break;

case 2:

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

scanf("%d", &data);

insertAtEnd(data);

break;

case 3:

printf("Enter data to delete: ");

scanf("%d", &data);

deleteNode(data);

break;

case 4:

printf("Enter data to search: ");

scanf("%d", &data);

search(data);

break;

case 5:

display();

break;

case 6:

printf("Exiting program\n");

exit(0);

default:

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

}

}

return 0;

}