Create, delete and display Linked List

#include<stdio.h>

#include<stdlib.h>

struct Node {

int data;

struct Node \*link;

};

typedef struct Node node;

node \*new1, \*curr, \*start = NULL;

void create();

void display();

void delete();

void delete\_beg();

void delete\_specified();

void delete\_end();

int main() {

int ch;

while(1) {

printf("1. Create 2. Delete 3. Display 4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &ch);

switch(ch) {

case 1:

create();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(0);

default:

printf("Wrong choice\n");

}

}

return 0;

}

void create() {

char ch;

do {

new1 = (node\*)malloc(sizeof(node));

printf("Enter value: \n");

scanf("%d", &new1->data);

if(start == NULL) {

start = new1;

curr = new1;

} else {

curr->link = new1;

curr = new1;

}

printf("Do you want to add another element? (Y/N)\n");

getchar();

scanf("%c", &ch);

} while(ch == 'y' || ch == 'Y');

curr->link = NULL;

}

void display() {

printf("Ruqaiyya Mahreen 1BM23EE044\n");

node \*temp;

if(start == NULL) {

printf("Linked list is empty\n");

return;

}

printf("Elements of list: ");

temp = start;

while(temp != NULL) {

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

temp = temp->link;

}

printf("NULL\n");

}

void delete() {

int choice;

printf("1. Delete from First Position\n");

printf("2. Delete a Specific Element\n");

printf("3. Delete from Last Position\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

delete\_beg();

break;

case 2:

delete\_specified();

break;

case 3:

delete\_end();

break;

default:

printf("Wrong choice\n");

}

}

void delete\_beg() {

if(start == NULL) {

printf("List is empty. No element to delete.\n");

return;

}

node \*temp = start;

start = start->link; // Point to the second node

printf("Deleted element: %d\n", temp->data);

free(temp); // Free memory of the first node

}

void delete\_end() {

if(start == NULL) {

printf("List is empty. No element to delete.\n");

return;

}

node \*temp = start;

if(temp->link == NULL) { // Only one node

printf("Deleted element: %d\n", temp->data);

free(temp);

start = NULL;

return;

}

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

temp = temp->link; // Traverse to the second last node

}

node \*last = temp->link;

temp->link = NULL; // Unlink the last node

printf("Deleted element: %d\n", last->data);

free(last);

}

void delete\_specified() {

int value;

printf("Enter value to delete: ");

scanf("%d", &value);

if(start == NULL) {

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

return;

}

node \*temp = start, \*prev = NULL;

// If the node to delete is the first node

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

start = temp->link;

printf("Deleted element: %d\n", temp->data);

free(temp);

return;

}

// Search for the node to delete

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

prev = temp;

temp = temp->link;

}

// Node not found

if(temp == NULL) {

printf("Element not found in the list.\n");

return;

}

// Unlink the node and free the memory

prev->link = temp->link;

printf("Deleted element: %d\n", temp->data);

free(temp);

}

