//DS LAB: 9 – IMPLEMENTATION OF STACK USING LINKED LIST

#include <stdio.h>

#include <stdlib.h>

// Define a structure for a node in the linked list

struct Node {

int data;

struct Node\* next;

};

// Function to create a new node

struct Node\* createNode(int data) {

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

if (newNode == NULL) {

printf("Memory allocation failed\n");

exit(1);

}

newNode->data = data;

newNode->next = NULL;

return newNode;

}

// Function to display the linked list (stack)

void display(struct Node\* top) {

if (top == NULL) {

printf("Stack is empty\n");

return;

}

printf("Stack elements: ");

while (top != NULL) {

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

top = top->next;

}

printf("\n");

}

// Function to push an element onto the stack

struct Node\* push(struct Node\* top, int data) {

struct Node\* newNode = createNode(data);

newNode->next = top;

return newNode;

}

// Function to pop an element from the stack

struct Node\* pop(struct Node\* top) {

if (top == NULL) {

printf("Stack is empty\n");

return NULL;

}

struct Node\* temp = top;

top = top->next;

free(temp);

return top;

}

// Function to free the entire linked list (stack)

void freeStack(struct Node\* top) {

while (top != NULL) {

struct Node\* temp = top;

top = top->next;

free(temp);

}

}

int main() {

struct Node\* top = NULL;

int choice, data;

while (1) {

printf("\nStack Menu:\n");

printf("1. Push (Insert)\n");

printf("2. Pop (Delete)\n");

printf("3. Display\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter data to push onto the stack: ");

scanf("%d", &data);

top = push(top, data);

break;

case 2:

top = pop(top);

break;

case 3:

display(top);

break;

case 4:

freeStack(top); // Free memory before exiting

exit(0);

default:

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

}

}

return 0;

}

/\* Output :

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 1

Enter data to push onto the stack: 10

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 1

Enter data to push onto the stack: 22

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 1

Enter data to push onto the stack: 64

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 1

Enter data to push onto the stack: 80

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 2

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 3

Stack elements: 64 22 10

Stack Menu:

1. Push (Insert)

2. Pop (Delete)

3. Display

4. Exit

Enter your choice: 4

\*/