# **Implementation of Stack using Linked List**

Shivani S - 192471030

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node \*next;

};

struct Node \*top = NULL;

void push(int value);

void pop(void);

void display(void);

int main(void) {

int choice, value;

printf("\n:: Stack using Linked List ::\n");

while (1) {

printf("\n\*\*\*\*\*\* MENU \*\*\*\*\*\*\n");

printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");

printf("Enter your choice: ");

if (scanf("%d", &choice) != 1) return 0;

switch (choice) {

case 1:

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

if (scanf("%d", &value) != 1) return 0;

push(value);

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

exit(0);

default:

printf("\nWrong selection! Please try again!\n");

}

}

return 0;

}

void push(int value) {

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

if (!newNode) {

printf("\nMemory allocation failed!\n");

return;

}

newNode->data = value;

newNode->next = top;

top = newNode;

printf("\nInsertion successful!\n");

}

void pop(void) {

if (top == NULL) {

printf("\nStack is Empty!\n");

return;

}

struct Node \*temp = top;

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

top = temp->next;

free(temp);

}

void display(void) {

if (top == NULL) {

printf("\nStack is Empty!\n");

return;

}

struct Node \*temp = top;

printf("\nStack elements (top to bottom):\n");

while (temp) {

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

if (temp->next) printf(" -> ");

temp = temp->next;

}

printf(" -> NULL\n");

} }

printf(" -> NULL\n");

}

