#include <stdio.h>

#include <stdlib.h>

// Structure for a stack node

struct Node {

int data;

struct Node\* next;

};

struct Node\* top = NULL; // Top of stack

// Function to push an element into stack

void push(int value) {

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

if (!newNode) {

printf("Heap Overflow! Cannot push %d\n", value);

return;

}

newNode->data = value;

newNode->next = top; // New node points to previous top

top = newNode; // Top now becomes new node

printf("%d pushed into stack.\n", value);

}

// Function to pop an element from stack

void pop() {

if (top == NULL) {

printf("Stack Underflow! Cannot pop.\n");

return;

}

struct Node\* temp = top;

printf("%d popped from stack.\n", top->data);

top = top->next;

free(temp);

}

// Function to display stack elements

void display() {

if (top == NULL) {

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

return;

}

printf("Stack elements are:\n");

struct Node\* temp = top;

while (temp != NULL) {

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

temp = temp->next;

}

}

int main() {

int choice, value;

while (1) {

printf("\n--- Stack Menu ---\n");

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

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to push: ");

scanf("%d", &value);

push(value);

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

printf("Exiting...\n");

exit(0);

default:

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

}

}

}