#include <stdio.h>

#define MAX 10

#define TRUE 1

#define FALSE 0

int stack[MAX];

int top = -1;

int isFull() {

return (top == MAX - 1);

}

int isEmpty() {

return (top == -1);

}

void push(int value) {

if (isFull()) {

printf("Stack Overflow!\n");

} else {

stack[++top] = value;

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

}

}

int pop() {

10

if (isEmpty()) {

printf("Stack Underflow!\n");

return -1; // Return -1 indicating stack is empty

} else {

return stack[top--];

}

}

int checkPalindrome() {

int i;

for (i = 0; i <= top / 2; i++) {

if (stack[i] != stack[top - i]) {

return FALSE;

}

}

return TRUE;

}

void displayStack() {

if (isEmpty()) {

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

} else {

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

for (int i = top; i >= 0; i--) {

printf("%d\n", stack[i]);

}

}

}

int main() {

int choice, value;

printf("Stack Operations:\n");

while (1) {

printf("1. Push an element onto stack\n");

printf("2. Pop an element from stack\n");

printf("3. Check if stack contents are palindrome\n");

printf("4. Display stack status (Overflow or Underflow)\n");

printf("5. Display all elements of stack\n");

printf("6. Exit\n");

printf("Enter your choice (1-6): ");

scanf("%d", &choice);

switch (choice) {

11

case 1:

printf("Enter integer to push: ");

scanf("%d", &value);

push(value);

break;

case 2:

value = pop();

if (value != -1) {

printf("Popped element: %d\n", value);

}

break;

case 3:

if (checkPalindrome()) {

printf("Stack contents are a palindrome.\n");

} else {

printf("Stack contents are not a palindrome.\n");

}

break;

case 4:

if (isFull()) {

printf("Stack Overflow!\n");

} else if (isEmpty()) {

printf("Stack Underflow!\n");

} else {

printf("Stack is neither full nor empty.\n");

}

break;

case 5:

displayStack();

break;

case 6:

printf("Exiting program.\n");

return 0;

default:

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

}

printf("\n");

}

return 0;

}