

```
3..#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define MAX 100
// Stack structure
int stack[MAX];
int top = -1;
// Function Prototypes
void push(int);
int pop();
void display();
bool isOverflow();
bool isUnderflow();
void checkPalindrome();
int main() {
    int choice, element;
    while (1) {
        printf("\n--- STACK MENU ---\n");
        printf("1. Push an Element onto Stack\n");
        printf("2. Pop an Element from Stack\n");
        printf("3. Check if Stack is Palindrome\n");
        printf("4. Demonstrate Overflow and Underflow\n");
        printf("5. Display Stack Status\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice) {
            case 1:
                printf("Enter element to push: ");
                scanf("%d", &element);
                push(element);
                break;
            case 2:
                element = pop();
                if (element != -1)
                    printf("Popped Element: %d\n", element);
                break;
            case 3:
                checkPalindrome();
                break;
            case 4:
                if (isOverflow())
                    printf("Overflow: Stack is Full!\n");
                else if (isUnderflow())
```

```

        printf("Underflow: Stack is Empty!\n");
    else
        printf("No Overflow or Underflow. Stack is within limits.\n");
    break;
case 5:
    display();
    break;
case 6:
    exit(0);
default:
    printf("Invalid choice! Try again.\n");
}
}
return 0;
}

// Function to push element
void push(int value) {
    if (top >= MAX - 1) {
        printf("Stack Overflow! Cannot push %d\n", value);
        return;
    }
    stack[++top] = value;
    printf("%d pushed onto the stack.\n", value);
}

// Function to pop element
int pop() {
    if (top == -1) {
        printf("Stack Underflow! Cannot pop.\n");
        return -1;
    }
    return stack[top--];
}

// Display Stack contents
void display() {
    if (top == -1) {
        printf("Stack is Empty.\n");
        return;
    }
    printf("Stack (Top to Bottom): ");
    for (int i = top; i >= 0; i--)
        printf("%d ", stack[i]);
    printf("\n");
}

// Check for overflow

```

```
bool isOverflow() {
    return top == MAX - 1;
}
// Check for underflow
bool isUnderflow() {
    return top == -1;
}
// Function to check if the stack is a palindrome
void checkPalindrome() {
    if (top == -1) {
        printf("Stack is empty. Cannot check for palindrome.\n");
        return;
    }
    int isPalindrome = 1;
    for (int i = 0; i <= top / 2; i++) {
        if (stack[i] != stack[top - i]) {
            isPalindrome = 0;
            break;
        }
    }
    if (isPalindrome)
        printf("Stack content is a Palindrome.\n");
    else
        printf("Stack content is NOT a Palindrome.\n");
}
```