P3→

STACK of Integers (Array Implementation of Stack with maximum size MAX)

a) Push an Element on to Stack

b) Pop an Element from Stack

c) Demonstrate how Stack can be used to check Palindrome

d) Demonstrate Overflow and Underflow situations on Stack

e) Display the status of Stack

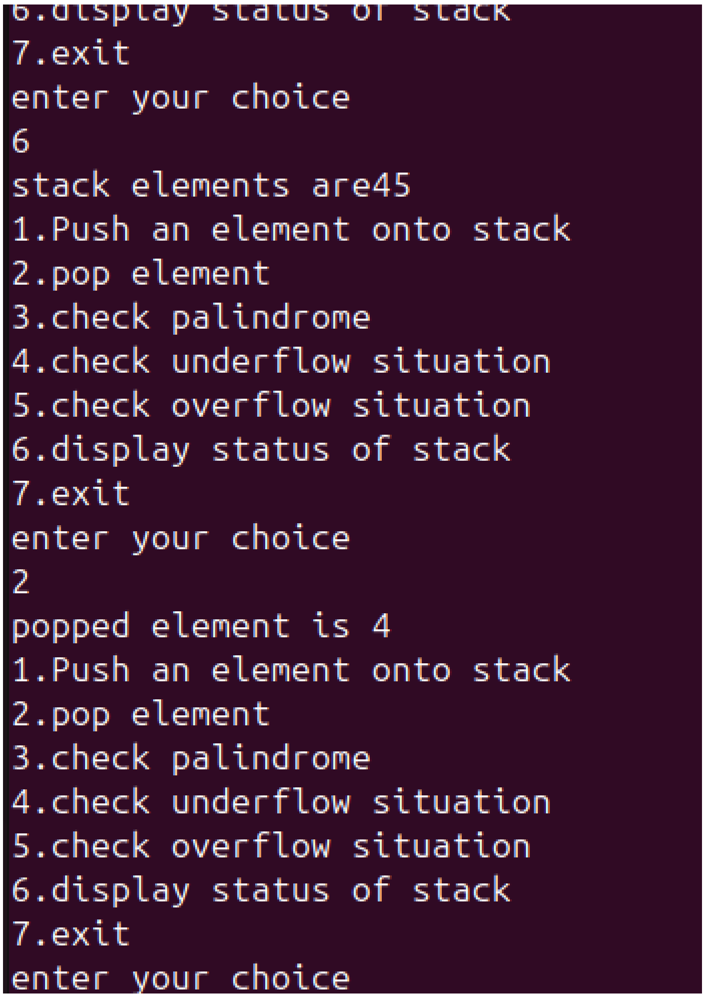
f) Exit

Support the program with appropriate functions for each of the above operations

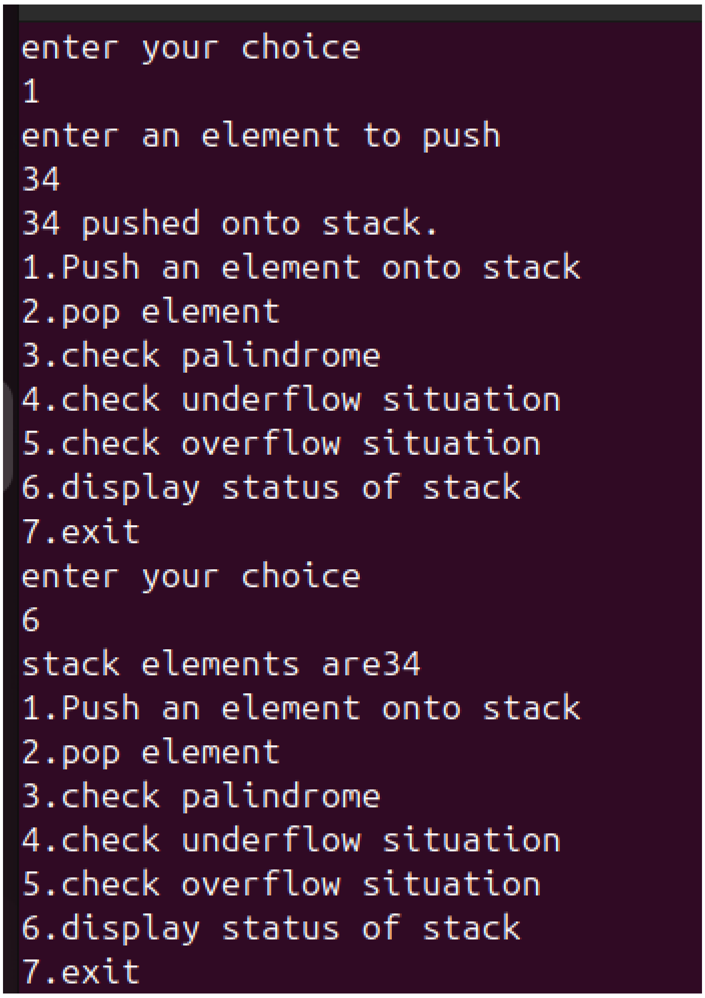
CODE=>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100

// Global stack and top pointer
int stack[MAX];
int top = -1;

// Function prototypes
void push(int elem);
int pop();
void display();
void checkOverflow();
void checkUnderflow();
void checkPalindrome();

int main() {
    int choice, elem;

    while (1) {
        printf("\n--- STACK MENU ---\n");
        printf("1. Push an Element\n");
        printf("2. Pop an Element\n");
        printf("3. Check Palindrome using Stack\n");
        printf("4. Demonstrate Overflow\n");
        printf("5. Demonstrate Underflow\n");
        printf("6. Display Stack\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter element to push: ");
                scanf("%d", &elem);
                push(elem);
                break;
            case 2:
                elem = pop();
                if (elem != -1)
                    printf("Popped element: %d\n", elem);
                break;
            case 3:
                checkPalindrome();
                break;
            case 4:
                checkOverflow();
                break;
            case 5:
                checkUnderflow();
                break;
            case 6:
                display();
                break;
            case 7:
                exit(0);
            default:
                printf("Invalid choice!\n");
        }
    }
    return 0;
}

// Push operation
void push(int elem) {
    if (top == MAX - 1) {
        printf("Stack Overflow! Cannot push %d\n", elem);
        return;
    }
    stack[++top] = elem;
    printf("%d pushed onto stack.\n", elem);
}

// Pop operation
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 elements: ");
    for (int i = top; i >= 0; i--) {
        printf("%d ", stack[i]);
    }
    printf("\n");
}

// Demonstrate overflow by trying to push beyond MAX
void checkOverflow() {
    int i;
    top = -1; // reset stack
    for (i = 0; i <= MAX; i++) {
        push(i);
    }
}

// Demonstrate underflow by trying to pop from empty stack
void checkUnderflow() {
    top = -1; // reset stack
    pop();
}

// Palindrome check using stack
void checkPalindrome() {
    char str[100];
    int i, len, flag = 1;
    int tempTop = -1;
    int tempStack[MAX];

    printf("Enter a string: ");
    scanf("%s", str);
    len = strlen(str);

    // push all characters
    for (i = 0; i < len; i++) {
        tempStack[++tempTop] = str[i];
    }

    // compare with pop
    for (i = 0; i < len; i++) {
        if (str[i] != tempStack[tempTop--]) {
            flag = 0;
            break;
        }
    }

    if (flag)
        printf("%s is a Palindrome.\n", str);
    else
        printf("%s is NOT a Palindrome.\n", str);
}
```



```
6.display status of stack
7.exit
enter your choice
6
stack elements are45
1.Push an element onto stack
2.pop element
3.check palindrome
4.check underflow situation
5.check overflow situation
6.display status of stack
7.exit
enter your choice
2
popped element is 4
1.Push an element onto stack
2.pop element
3.check palindrome
4.check underflow situation
5.check overflow situation
6.display status of stack
7.exit
enter your choice
```



```
enter your choice
1
enter an element to push
34
34 pushed onto stack.
1.Push an element onto stack
2.pop element
3.check palindrome
4.check underflow situation
5.check overflow situation
6.display status of stack
7.exit
enter your choice
6
stack elements are34
1.Push an element onto stack
2.pop element
3.check palindrome
4.check underflow situation
5.check overflow situation
6.display status of stack
7.exit
```