```
#include <stdio.h>
#include <ctype.h>   // for isdigit
#define MAX 100
int stack[MAX];
int top = -1;
// Push element onto stack
void push(int x) {
    if (top == MAX - 1) {
        printf("Stack Overflow\n");
    } else {
        stack[++top] = x;
    }
}
// Pop element from stack
int pop() {
    if (top == -1) {
        printf("Stack Underflow\n");
        return -1;
    } else {
        return stack[top--];
    }}
```

```
int evaluatePostfix(char postfix[]) {
    int i, op1, op2;
    char ch;
    for (i = 0; postfix[i] != '\0'; i++) {
        ch = postfix[i];
        // If operand (digit), push it
        if (isdigit(ch)) {
            push(ch - '0');  // convert char to int
        }
        // If operator, pop two operands and apply
        else {
            op2 = pop();
            op1 = pop();
            switch (ch) {
                case '+': push(op1 + op2); break;
                case '-': push(op1 - op2); break;
                case '*': push(op1 * op2); break;
                case '/': push(op1 / op2); break;
                case '^': {
                    int result = 1;
                    for (int j = 0; j < op2; j++)
```

```
                        result *= op1;
                    push(result);
                    break;
                }
                default:
                    printf("Invalid operator: %c\n", ch);
                    return -1;
            }
        }
    }

    return pop();  // Final result
}
int main() {
    char postfix[MAX];
    printf("Enter postfix expression: ");
    scanf("%s", postfix);
    int result = evaluatePostfix(postfix);
    printf("Result of postfix evaluation: %d\n", result);
    return 0;
}
```