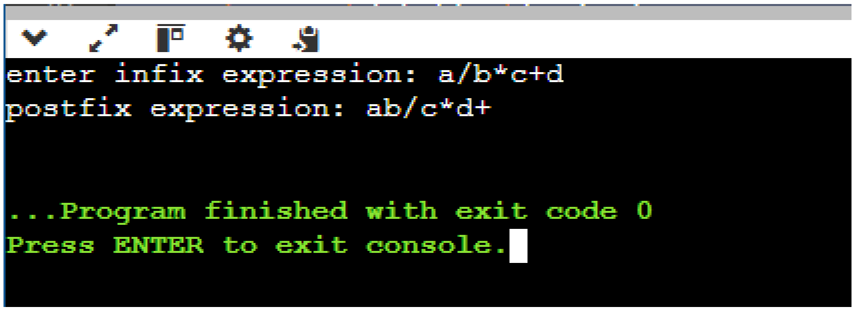
Name: Mithlesh Yeole
Section/Batch: B3/B3
Roll no.: 59
Date: 14th Feb 2024

Practical 2 (B)

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#define MAX 100
typedef struct {
    char data[MAX];
    int top;
} Stack;
void push(Stack *s, char item) {
    if (s->top == (MAX - 1)) {
        printf("Stack overflow\n");
        return;
    }
    s->data[++(s->top)] = item;
}
char pop(Stack *s) {
    if (s->top == -1) {
        printf("Stack underflow\n");
        exit(1);
    }
    return s->data[(s->top)--];
}
char peek(Stack *s) {
    if (s->top == -1) {
        printf("Stack is empty\n");
        exit(1);
    }
    return s->data[s->top];
}
int precedence(char op) {
    switch (op) {
        case '+':
        case '-':
            return 1;
        case '*':
        case '/':
            return 2;
        case '^':
            return 3;
        default:
            return 0;
    }
}
int isOperator(char ch) {
    return ch == '+' || ch == '-' || ch == '*' || ch == '/' || ch == '^';
}
void infixtoPostfix(char infix[], char postfix[]) {
    Stack s;
    s.top = -1;
    int i, j = 0;
    char item;
    for (i = 0; infix[i] != '\0'; i++) {
        item = infix[i];
        if (isdigit(item) || isalpha(item)) {
            postfix[j++] = item;
        } else if (item == '(') {
            push(&s, item);
        } else if (item == ')') {
            while (s.top != -1 && peek(&s) != '(') {
                postfix[j++] = pop(&s);
            }
            if (s.top != -1) {
                pop(&s);
            }
        } else if (isOperator(item)) {
            while (s.top != -1 && precedence(peek(&s)) >= precedence(item)) {
                postfix[j++] = pop(&s);
            }
            push(&s, item);
        }
    }
    while (s.top != -1) {
        postfix[j++] = pop(&s);
    }
    postfix[j] = '\0';
}
int main() {
    char infix[MAX], postfix[MAX];
    printf("enter infix expression: ");
    scanf("%s", infix);
    infixtoPostfix(infix, postfix);
    printf("postfix expression: %s\n", postfix);
    return 0;
}
```



```
enter infix expression: a/b*c+d
postfix expression: ab/c*d+


...Program finished with exit code 0
Press ENTER to exit console.
```