**CC LAB 07** | Postfix Evaluation

**Aim:** Lex program to evaluate a postfix expression

**Implementation:**

%{

*#include* <stdio.h>

*#include* <stdlib.h>

*#define* MAX\_STACK\_SIZE 100

int stack[MAX\_STACK\_SIZE];

int top = -1;

void push(int *value*) {

*if* (top >= MAX\_STACK\_SIZE - 1) {

        fprintf(stderr, "Stack overflow\n");

        exit(1);

    }

    stack[++top] = value;

}

int pop() {

*if* (top < 0) {

        fprintf(stderr, "Stack underflow\n");

        exit(1);

    }

*return* stack[top--];

}

%}

%option noyywrap

%%

[0-9]+     { int value = atoi(yytext); push(value); }

[-+\*/]     {

              int operand2 = pop();

              int operand1 = pop();

              int result;

*switch* (yytext[0]) {

*case* '+': result = operand1 + operand2; *break*;

*case* '-': result = operand1 - operand2; *break*;

*case* '\*': result = operand1 \* operand2; *break*;

*case* '/': result = operand1 / operand2; *break*;

              }

              push(result);

          }

[ \t\n]    ; *// Ignore whitespace*

.          {

              fprintf(stderr, "Invalid character: %s\n", yytext);

              exit(1);

          }

%%

int main(int *argc*, char \**argv*[]) {

*if* (argc==2) {

        yyin=fopen(argv[1],"r");

    }

*else* {

        printf("\nEnter the input:\n");

        yyin=stdin;

    }

    yylex();

*if* (top != 0) {

        fprintf(stderr, "Invalid expression\n");

*return* 1;

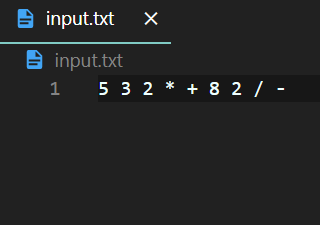
    }

    printf("Result: %d\n", stack[top]);

*return* 0;

}

**Input:**

****

**Output:**

