**CSA1443- Compiler Design for Intraprocedural Analysis**

**192321047 – KAUSHIK NARAYANAN.V**

**15. Write a C Program to implement Operator Precedence Parsing.**

**Aim:**  
To implement Operator Precedence Parsing in C to parse and evaluate a given input expression using operator precedence relations.

**Code:**

#include <stdio.h>

#include <string.h>

int prec(char c) {

if (c == '+' || c == '-') return 1;

if (c == '\*' || c == '/') return 2;

return 0;

}

void parse(char \*exp) {

char stack[50];

int top = -1, i = 0;

stack[++top] = '$'

while (exp[i] != '\0') {

while (top >= 0 && prec(stack[top]) >= prec(exp[i])) {

printf("Reduce by popping %c\n", stack[top--]);

}

printf("Shift '%c'\n", exp[i]);

stack[++top] = exp[i++];

}

while (top > 0) {

printf("Reduce by popping %c\n", stack[top--]);

}

printf("Parsing Successful!\n");

}

int main() {

char exp[50];

printf("Enter expression: ");

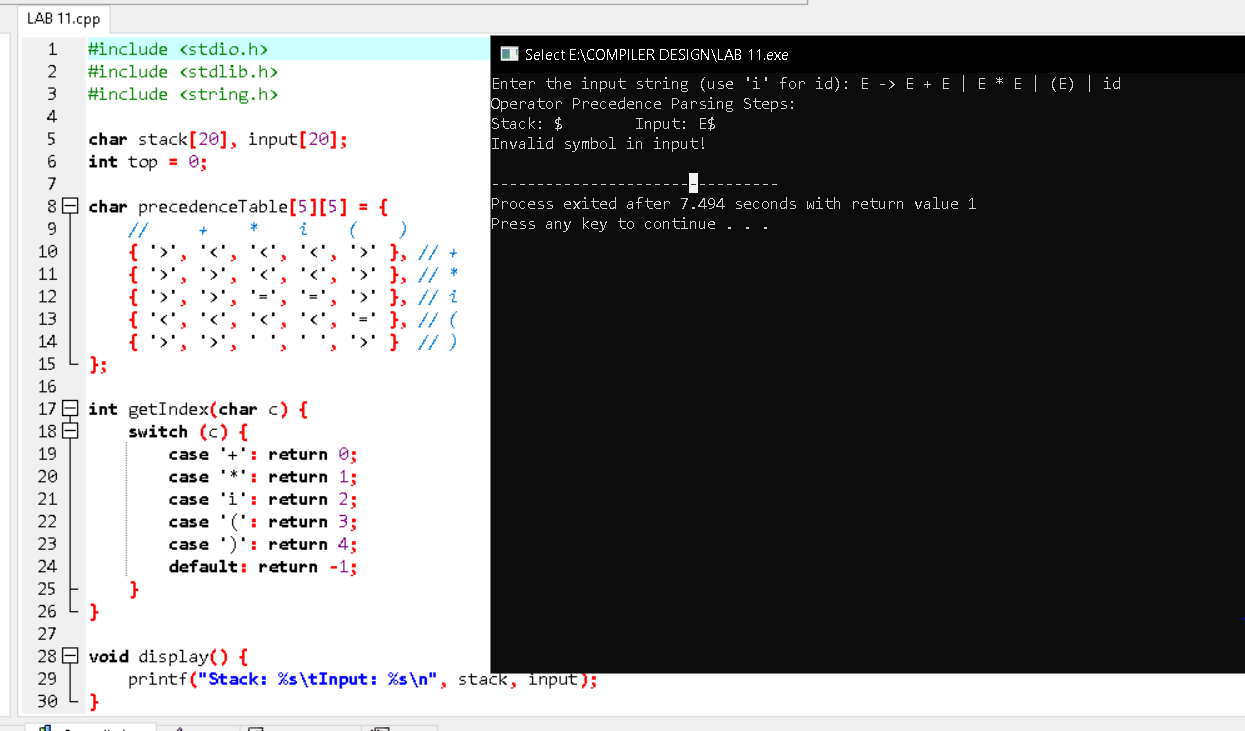
scanf("%s", exp);

parse(exp);

return 0;

}

**Output:**

****

**16. Write a C Program to Generate the Three-Address Code representation for the given input statement.**

**Aim:**  
To implement a C program that generates three-address code for arithmetic expressions.

**Code:**

#include <stdio.h>

#include <string.h>

int main() {

char expr[50], temp[5] = "t";

int i = 0, j = 1, k = 0;

printf("Enter an arithmetic expression: ");

scanf("%s", expr);

printf("\nThree-Address Code:\n");

while (expr[i] != '\0') {

if (expr[i] == '+' || expr[i] == '-' || expr[i] == '\*' || expr[i] == '/') {

printf("t%d = %c %c %c\n", k++, expr[i - 1], expr[i], expr[i + 1]);

i++;

}

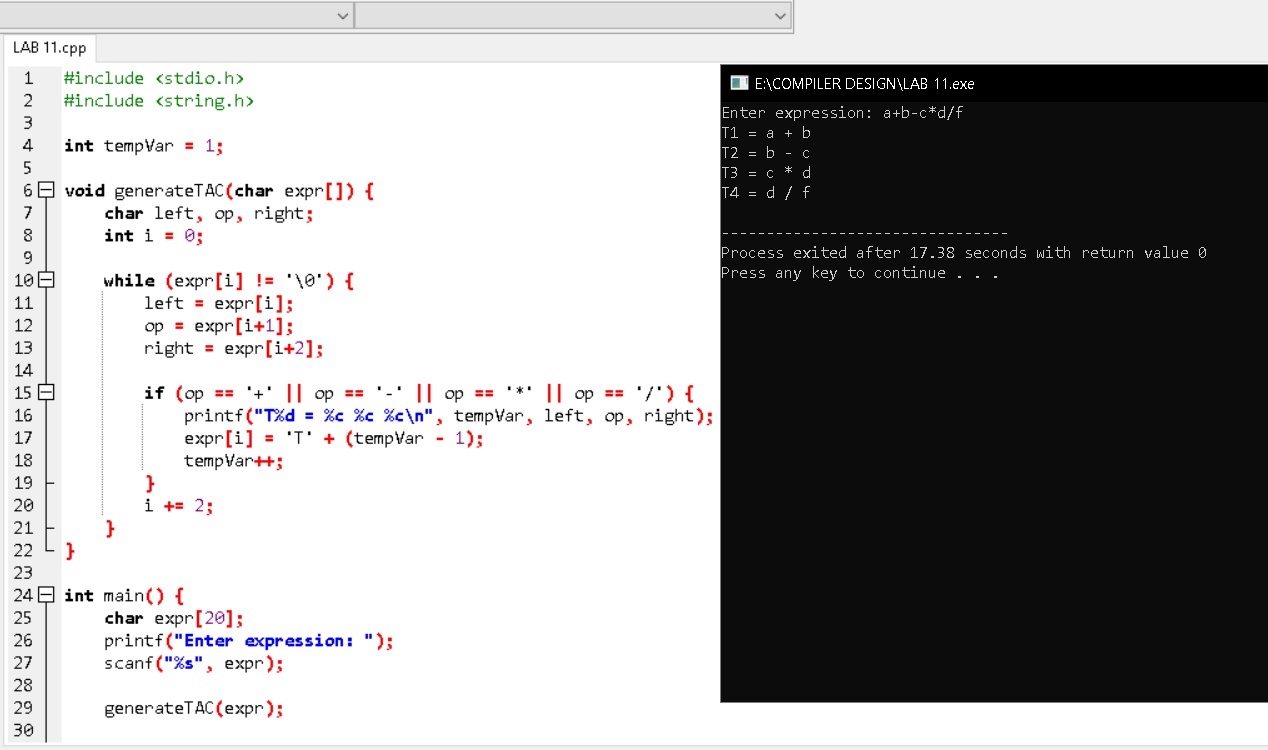
i++;

}

return 0;

}

**Output:**

****