TOPIC NO.: 7

TITLE: Infix to Postfix

Program Code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#define MAX\_SIZE 100

int isOperator(char ch) {

return (ch == '+' || ch == '-' || ch == '\*' || ch == '/');

}

int getPrecedence(char op) {

if (op == '+' || op == '-')

return 1;

if (op == '\*' || op == '/')

return 2;

return 0;

}

void infixToPostfix(char infix[], char postfix[]) {

int i, j;

char stack[MAX\_SIZE];

int top = -1;

for (i = 0, j = 0; infix[i] != '\0'; i++) {

if (infix[i] == ' ')

continue;

else if (isalnum(infix[i]))

postfix[j++] = infix[i];

else if (infix[i] == '(')

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

Output:

Enter the infix expression: a+b\*(c/d-(e\*f)+h)

Postfix expression: abcd/ef\*-h+\*+

DATE: 28.02.24

else if (infix[i] == ')') {

while (top != -1 && stack[top] != '(') {

postfix[j++] = stack[top--];

}

top--; // Pop '('

}

else if (isOperator(infix[i])) {

while (top != -1 && getPrecedence(stack[top]) >= getPrecedence(infix[i])) {

postfix[j++] = stack[top--];

}

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

}

}

while (top != -1) {

postfix[j++] = stack[top--];

}

postfix[j] = '\0'; // Null terminate the expression

}

int main() {

char infix[MAX\_SIZE];

char postfix[MAX\_SIZE];

printf("Enter the infix expression: ");

fgets(infix, MAX\_SIZE, stdin);

infixToPostfix(infix, postfix);

printf("Postfix expression: %s", postfix);

return 0;

}