Ex. No. 4c **Postfix Expression Evaluation**

Date:

**Aim**

To evaluate the given postfix expression using stack operations.

**Algorithm**

1. Start

2. Define a array stack of size max = 20

3. Initialize top = -1

4. Read the postfix expression character-by-character If character is an operand push it onto the stack If character is an operator Pop topmost two elements from stack. Apply operator on the elements and push the result onto the stack,

5. Eventually only result will be in the stack at end of the expression.

6. Pop the result and print it.

7. Stop

**Program**

/\* Evaluation of Postfix expression using stack \*/

#include <stdio.h>

#include <stdlib.h>

struct stack

{

int top;

float a[50];

}s;

void main()

{

char pf[50];

float d1,d2,d3;

int i;

system("clear");

s.top = -1;

printf("\n\n Enter the postfix expression: ");

fgets(pf,20,stdin);

for(i=0; pf[i]!='\0'; i++)

{

switch(pf[i])

{

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

s.a[++s.top] = pf[i]-'0';

break;

case '+':

d1 = s.a[s.top--];

d2 = s.a[s.top--];

s.a[++s.top] = d1 + d2;

break;

case '-':

d2 = s.a[s.top--];

d1 = s.a[s.top--];

s.a[++s.top] = d1 - d2;

break;

case '\*':

d2 = s.a[s.top--];

d1 = s.a[s.top--];

s.a[++s.top] = d1\*d2;

break;

case '/':

d2 = s.a[s.top--];

d1 = s.a[s.top--];

s.a[++s.top] = d1 / d2;

break;

}

}

printf("\n Expression value is %5.2f", s.a[s.top]);

}

**Output**

**Result**

Thus the given postfix expression was evaluated using stack.