AIM:

Write a C program to evaluate Arithmetic expression using stack

ALGORITHM

 Make an empty stack.

 Read the postfix expression one character at a time until it encounters end of

expression.

 If the character is an operand, push its associated value onto the stack.

 If the character is an operator, pop two values from the stack, apply the operator to

them and push the result onto the stack.

PROGRAM

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX 20

struct node

{

int Element;

struct node \*Next;

} \*List = NULL;

typedef struct node Stack;

void Push(int e);

int Pop();

int main()

{

int i, a, b, c, e;

char expr[MAX];

printf("Enter the postfix expression : ");

gets(expr);

for(i = 0; i < strlen(expr); i++)

{

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

{

b = Pop();

a = Pop();

switch(expr[i])

{

case '+':

c = a + b;

Push(c);

break;

case '-':

c = a - b;

Push(c);

break;

case '\*':

c = a \* b;

Push(c);

break;

case '/':

c = a / b;

Push(c);

break;

}

}

else

{

printf("Enter the value of %c : ", expr[i]);

scanf("%d", &e);

Push(e);

}

}

printf("The result is %d", Pop());

return 0;

}

void Push(int e)

{

Stack \*NewNode = malloc(sizeof(Stack));

NewNode->Element = e;

if(List == NULL)

NewNode->Next = NULL;

else

NewNode->Next = List;

List = NewNode;

}

int Pop()

{

int e;

Stack \*TempNode;

TempNode = List;

List = List->Next;

e = TempNode->Element;

free(TempNode);

return e;

}

OUTPUT

Enter the postfix expression : abc+\*d\*

Enter the value of a : 2

Enter the value of b : 3

Enter the value of c : 4

Enter the value of d : 5

The result is 70

RESULT

Hence a C program to evaluate Arithmetic expression using stack is implemented.