Name : Devvrat Miglani

Section : E Rollno : 37

Batch:E3

**Practical No. 2: Stack ADT Implementation**

Q1) To Study and implement Stack ADT and write function to evaluate Postfix Expression

using Stack.

Input:- 5\*4+2

Output: 1. Postfix Expression : 54\*2+

2. Postfix Evaluation : 22

Code:

#include<stdio.h>

#include<ctype.h>

#include<string.h>

int max = 5;

int stack[10];

int top = -1;

int isfull(){

if(top == max){

return 1;

}

else{

return 0;

}

}

int isempty(){

if(top == -1){

return 1;

}

else{

return 0;

}

}

void push(int data){

if(!isfull()){

top = top + 1;

stack[top] = data;

}

else{

printf("The sttack is full");

}

}

int pop(){

int data;

if(!isempty()){

data = stack[top];

top = top - 1;

}

else{

printf("stack is empty");

}

return data;

}

int priority(char x)

{

if(x == '(')

return 0;

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

return 1;

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

return 2;

return 0;

}

int evalPostfix(char\* exp){

stack[strlen(exp)];

int i,a,b;

for (int i = 0; i < exp[i]; i++)

{

if (stack[i]>=0 && stack[i]<='9')

{

push(stack[i]-'0');

}else{

if (!isempty())

{

a = pop();

b = pop();

}

switch(stack[i]){

case '+':

push(b+a);

break;

case '-':

push(b-a);

break;

case '\*':

push(b\*a);

break;

case '/':

push(b/a);

break;

}

}

}

return pop();

}

void main(){

char exp[100];

char \*e, x;

printf("Enter the expression : ");

scanf("%s",exp);

printf("\n");

e = exp;

while(\*e != '\0')

{

if(isalnum(\*e))

printf("%c ",\*e);

else if(\*e == '(')

push(\*e);

else if(\*e == ')')

{

while((x = pop()) != '(')

printf("%c ", x);

}

else

{

while(priority(stack[top]) >= priority(\*e))

printf("%c ",pop());

push(\*e);

}

e++;

}

while(top != -1)

{

printf("%c \n",pop());

}

printf("Answer : %d",evalPostfix(exp));

}

Output:

