AIM: Evaluate Postfix Expression using Stack ADT.

CODE:

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<ctype.h>

#define MAX 100

int stack[MAX];

int top = -1;

void push(int item)

{

if (top >= MAX - 1)

{

printf("Stack overflow");

return;

}

else

{

top = top + 1;

stack[top] = item;

}

}

int pop()

{

int item;

if (top < 0)

{

printf("Stack underflow");

}

else

{

item = stack[top];

top = top - 1;

return item;

}

}

void evalPostfix(char postfix[])

{

int i;

char ch;

int val;

int A, B;

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

{

ch = postfix[i];

if (isdigit(ch))

{

push(ch - '0');

}

else if (ch == '+' || ch == '-' || ch == '\*' || ch == '/' || ch == '^')

{

A = pop();

B = pop();

switch (ch)

{

case '\*':

val = B \* A;

break;

case '/':

val = B / A;

break;

case '+':

val = B + A;

break;

case '-':

val = B - A;

break;

case '^':

val = pow(B , A);

break;

}

push(val);

}

}

printf("\nResult: %d \n", pop());

}

void main()

{

int i;

char postfix[MAX];

clrscr();

printf("\nEnter postfix expression,\nPress right parenthesis ')' for end expression: ");

for(i = 0; i <= MAX - 1; i++)

{

scanf("%c", &postfix[i]);

if(postfix[i]==')')

{

break;

}

}

evalPostfix(postfix);

getch();

}

OUTPUT:

