AIM: Write a program to evaluate a postfix expression

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<math.h>

/* Let the max value of the arrays we use be 100 */
#define MAX 100

/* Declare few global variables which can be used by more than 1 function */
int stack[MAX], top=-1;

/* We need the following operations to perform PostFix Evaluation
1) PUSH Function
2) POP Function
3) IsOperand Function
4) IsOperator Function
5) Operation Function (Optional, can also be included in the main())
*/

/* We don't consider the case to print stack overflow condition or underflow condition, as this will be printed in the output along with the Postfix expression.*/

/*------PUSH Function-----*/
void push(int value)
{
  ++top;
  stack[top]=value;
}
/*------POP Function------*/
int pop()
{
 int x = stack[top];
 --top;
 return(x);
}
/*------IsOperand Function---*/
/*This function returns 1(if operand) else 0*/
int isOperand(char symbol)
{
 if(symbol>='a' && symbol<='z')
    return 1;
 else
    return 0;
}
/*-----IsOperator Function ---*/
/*This Function returns 1(if operator) else returns 0*/
int isOperator(char symbol)
{
 if(symbol=='^' || symbol=='*' || symbol=='/' || symbol=='+' || symbol=='-')
    return 1;
 else
    return 0;
}
```

```
/*---------MAIN Function------*/
void main()
{
/*
STEP 1: Get the postfix expression
STEP 2: Evaluate each postfix symbol, if Operator or Operand
    2.1: If Operand, ask user for the value of operand
    2.2: 1: If Operator, pop first 2 elements from the postfix
      2.2.1: Perform the operation between the 2 operands
      2.2.2: Store it back into the stack
STEP 3: Repeat Step 2 till the end of Expression.
STEP 4: display final Result.
*/
  char postfix[MAX],symbol;
  int i=0, result, opr1, opr2, x, res_final;
  clrscr();
  printf("\nEnter the postfix Expression: ");
  gets(postfix);
  while(postfix[i]!='\0')
  {
    symbol=postfix[i];
    if(isOperand(symbol)==1)
    {
      printf("\nEnter the value of %c: ", symbol);
      scanf("%d", &x);
      push(x);
    }
    if(isOperator(symbol)==1)
    {
      opr2=pop();
      opr1=pop();
      switch(symbol)
      {
       case '^': result=pow(opr1,opr2);break;
       case '*': result=opr1*opr2;break;
       case '/': result=opr1/opr2;break;
       case '+':  result=opr1+opr2; break;
       case '-':  result=opr1-opr2; break;
      }
      push(result);
    }
    i++;//incrementing to check for another postfix symbol.
  }//While loop ending
  /*Once all the symbols are scanned, we pop the final result*/
   res_final=pop();
  printf("\nThe result of the Postfix Expression is: %d", res_final);
  getch();
}
```

OUTPUT

```
Enter the postfix Expression:
abc*+
Enter the value of a: 1
Enter the value of b: 2
Enter the value of c: 3
The result of the Postfix Expression is: 7
```