**DS Lab Program - 04**

Develop a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +,-,\*,/,%(Remainder), ^ (Power) and alphanumeric operands.

#include<stdio.h>

#include<ctype.h>

#define SIZE 50 //size of stack

//char elem;

char s[SIZE];

int top = -1; //global declarations

int push(char elem) //push() for push operation

{

s[++top] = elem;

}

char pop() //pop() for pop operation

{

return(s[top--]);

}

int pr(char elem) //pr() for precedence

{

switch(elem)

{

case '#' : return 0;

case '(' : return 1;

case '+' :

case '-' : return 2;

case '\*' :

case '/' :

case '%' : return 3;

case '^' : return 4;

}

}

void main()

{

char infx[50], pofx[50], ch, elem;

int i=0, k=0;

printf("\n\n Enter the valid Infix Expression");

scanf("%s", &infx);

push('#');

while((ch = infx[i++]) != '\0')

{

if(ch == '(')

{

push(ch);

}

else if(isalnum(ch))

{

pofx[k++] = ch;

}

else if(ch == ')')

{

while(s[top] != '(')

{

pofx[k++] = pop();

}

elem = pop();

}

else

{

while(pr(s[top]) >= pr(ch))

{

pofx[k++] = pop();

}

push(ch);

}

}

while(s[top] != '#') // pop from stack till empty

{

pofx[k++] = pop();

}

pofx[k] = '\0'; //make pofx as valid string

printf("\n\n Given Infix Expression: %s Postfix Expression: %s\n",infx,pofx);

}

