DATE : 20.12.2022

USN : 1NT21IS117

Q.Write a Program to convert infix expression to prefix expression.

AIM : Convert Infix to Prefix.

Algorithm :

Function InfixtoPrefix( stack, infix)

infix = reverse(infix)

loop i = 0 to infix.length

**if** infix[i] is operand → prefix+= infix[i]

**else** **if** infix[i] is '('  → stack.push(infix[i])

**else** **if** infix[i] is ')' → pop and print the values of stack till the symbol ')' is not found

**else** **if** infix[i] is an operator(+, -, \*, /, ^) →

**if** the stack is empty then push infix[i] on the top of the stack.

Else →

If precedence(infix[i] > precedence(stack.top))

→   Push infix[i] on the top of the stack

**else** **if**(infix[i] == precedence(stack.top) && infix[i] == '^')

→   Pop and print the top values of the stack till the condition is **true**

→   Push infix[i] into the stack

**else** **if**(infix[i] == precedence(stack.top))

→   Push infix[i] on to the stack

Else **if**(infix[i] < precedence(stack.top))

→   Pop the stack values and print them till the stack is not empty and infix[i] < precedence(stack.top)

→   Push infix[i] on to the stack

End loop

Pop and print the remaining elements of the stack

Prefix = reverse(prefix)

**return**

CODE :

#include<stdio.h>

#define max 100

int top=-1, a[max];

void push(char x)

{

a[++top]=x;

}

char pop()

{ if(top==-1)

return -1;

else

return a[top--];

}

int prcd(char c)

{ if(c==')')

return 0;

else if(c=='+'||c=='-')

return 1;

else if(c=='\*'||c=='/')

return 2;

}

void infixtoprefix(char infix[max],char prefix[max])

{

char temp,x;

int i=0,j=0;

strrev(infix);

while(infix[i]!='\0')

{

temp=infix[i];

if(isalnum(temp))

{

prefix[j++]=temp;

}

else if(temp==')')

push(temp);

else if(temp=='(')

{

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

{

prefix[j++]=x;

}

}

else

{ while(prcd(a[top])>=prcd(temp))

{prefix[j++]=pop();}

push(temp);

}

i++;

}

while(top!= -1)

prefix[j++]=pop();

prefix[j]='\0';

strrev(prefix);

}

main()

{

char infix[max],prefix[max];

printf("Enter the infix expression\n");

gets(infix);

printf("The infix expression is %s\n",infix);

infixtoprefix(infix, prefix);

printf("The prefix expression is %s\n",prefix);

}

OUTPUT :

