Program to convert infix to postfix

Program:-

//Infix to postfix//

#include<stdio.h>//standard input output header file//

#include<ctype.h>

char stack[20];

int top=-1;//declaring the value of top is -1//

void push(char x)//push function//

{

stack[++top]=x;

}

char pop()// pop function//

{

if(top==-1)

return -1;

else

return stack[top--];

}

if(isalnum(\*e))

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

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

push(\*e);//checks the condition of else if and pushes the element//

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)

{

e++;//incrementation of e by 1//

}

while(top!=-1)

{

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

}return 0;

}

Algorithm:-

Step 1: Scan the infix expression from left to right.

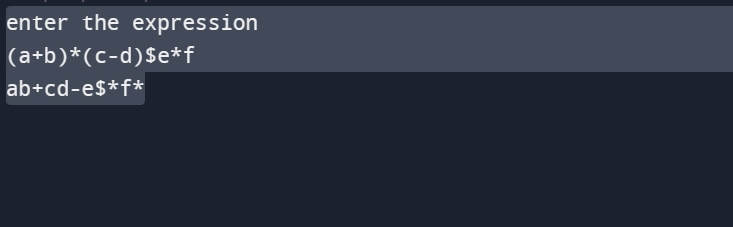
Step 2: If the second character is an operand, append it with final infix to the postfix.

Step 3: Else, if the precedence order of the scanned operator is greater than the precedence order of the operator in the stack.

Step 4: Else, pop all the operators from the stack which are greater than or equal to precedence than that of the scanned operator.Push it into the stack.

Step 5: If the scanned character is a ‘(‘, push it to the stack.

Step 6: Repeat the steps until infix expression is scanned.



[data-structures-2/21\_12\_22\_1NT21IS102\_PROGRAM3.docx at main · Niharika-PS/data-structures-2 (github.com)](https://github.com/Niharika-PS/data-structures-2/blob/main/21_12_22_1NT21IS102_PROGRAM3.docx)