Varsha. S

Date 5/10/2020

Expt. No. 1BM19CS179 Page No.

Lab program - 3

Algorithm to convert an infix to postfix notation.

Step 1 - Add ")" to the end of the infix expression.

Step 2 - Push "(" on to the stack.

Step 3 - Repeat until each character in the infix notation is scanned.

IF a "(" is encountered, push it on the stack. IF an operand (whether a digit or a character) is encountered, add it to the postfix expression

IF a ")" is encountered, then

a) Repeatedly pop from stack and add it to the postfix expression until a "(" is encountered

b) Discard the "(". That is, remove the "(" from stack and do not add it to postfix expression.

IF an operator O is encountered then.

Teacher's Signature :

a) Repeatedly pop from stack and add each operator (popped from the stack) to the postfix expression which has the same precedence or a higher precedence than O

b) Push the operator O to the stack.
(END OF IF)

Step 4 - Repeatedly pop from the stack and add it to the postfix expression until the stack is empty.

Step 5 - EXIT

Varsha.S

Date 5/10/2020

Expt. No. 1BM19CS179 Page No.

Lap program 3

Pseudocode.

```
Convert_infix_postfix (exp).
{
Create a stack S
for i=0 to length(exp) -1
{
{
if exp[i] is operand.
res = res + exp[1]
else
if exp[i] is operator
while ( is empty() && hashigherp
exp[i])
{
{
res = res + Stop( )
  S.pop
}
S. push [exp[i])
else if is opening par (exp[i])
 S.push (exp[i])
else if is closing par (exp[i])
while ( is empty() && is closingpar (
```

Teacher's Signature :

```
{
res= res + s.top( )
s.pop( )
}
}
}
while ( !s.empty( ))
{
res = res + s.top( )
s.pop( )
}
return res;
}
```