EXPERIMENT NO 10

```
OPERATORS = set(['+', '-', '*', '/', '(', ')'])
PRI = {'+':1, '-':1, '*':2, '/':2}
 def
infix_to_postfix(formula):
    stack = []      output =
''     for ch in formula:
if ch not in OPERATORS:
            output    +=    ch
elif ch == '(':
            stack.append('(')         elif
ch == ')':              while stack and
stack[-1] != '(':
                output += stack.pop()             stack.pop() # pop '('
else:             while stack and stack[-1] != '(' and PRI[ch] <=
PRI[stack[-1]]:
                output += stack.pop()
stack.append(ch)
     while
stack:
        output += stack.pop()
print(f'POSTFIX: {output}')
return output
 def
infix_to_prefix(formula):
    op_stack = []
exp_stack = []     for
ch in formula:
        if not ch in OPERATORS:
exp_stack.append(ch)         elif ch
== '(':
            op_stack.append(ch)
elif ch == ')':             while
op_stack[-1] != '(':                    op =
op_stack.pop()                  a =
exp_stack.pop()                  b =
exp_stack.pop()
exp_stack.append( op+b+a )
op_stack.pop() # pop '('
```

```
        else:
            while op_stack and op_stack[1]
!= '(' and PRI[ch] <= PRI[op_stack[-1]]:
                op = op_stack.pop()
a = exp_stack.pop()                 b =
exp_stack.pop()
exp_stack.append( op+b+a )
op_stack.append(ch)
         while
op_stack:
        op = op_stack.pop()
a = exp_stack.pop()         b =
exp_stack.pop()
exp_stack.append( op+b+a )
print(f'PREFIX: {exp_stack[-1]}')
return exp_stack[-1]


def generate3AC(pos):
exp_stack = []     t = 1
for i in pos:         if i not
in OPERATORS:
exp_stack.append(i)
else:
            print(f't{t} := {exp_stack[-2]} {i} {exp_stack[-1]}')
exp_stack=exp_stack[:-2]             exp_stack.append(f't{t}')
t+=1              def generate3ACTable(pos):
    exp_stack = []     t = 1
for i in pos:         if i not
in OPERATORS:
exp_stack.append(i)
else:
            print(f' {i}\t|\t{exp_stack[-2]}\t|\t{exp_stack[-1]}\t|\tt{t}
')             exp_stack=exp_stack[:-2]
exp_stack.append(f't{t}')             t+=1
 expres = input("INPUT THE EXPRESSION:
") pre = infix_to_prefix(expres)
print("") pos =
infix_to_postfix(expres)
generate3AC(pos)
```

```
print("\n-----------Quadruple Table----------------\n")
print("op\t|\targ1\t|\targ2\t|\tResult\n")
generate3ACTable(pos)
```

OUTPUT: