

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:

The screenshot shows a Python IDE with a file named `ThreeAC.py`. The code defines functions for infix to prefix, infix to postfix, and generating a 3-act table. The terminal output shows the program running with the input expression `(a+b*c)/(d-e)*f`. It displays the postfix expression `abc*+de-/f*` and the resulting quadruple table.

```
PS C:\Users\jinda\Downloads> & 'c:\Users\jinda\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\jinda\.vscode\extensions\ms-python.python-2021.5.829140558\pythonFiles\lib\python\debugpy\launcher' '51436' '-...' 'c:\Users\jinda\Downloads\ThreeAC.py'
INPUT THE EXPRESSION: (a+b*c)/(d-e)*f
PREFIX: */+*bc-def

POSTFIX: abc*+de-/f*
t1 := b * c
t2 := a + t1
t3 := d - e
t4 := t2 / t3
t5 := t4 * f

-----Quadruple Table-----

op |      arg1      |      arg2      |      Result
*   |      b         |      c         |      t1
+   |      a         |      t1        |      t2
-   |      d         |      e         |      t3
/   |      t2        |      t3        |      t4
*   |      t4        |      f         |      t5

PS C:\Users\jinda\Downloads>
```

The screenshot shows the same Python IDE with the `ThreeAC.py` file. The terminal output shows the program running with the input expression `a+b*c*d/e`. It displays the postfix expression `ab+cd*e/-` and the resulting quadruple table.

```
n.python-2021.5.829140558\pythonFiles\lib\python\debugpy\launcher' '51449' '-...' 'c:\Users\jinda\Downloads\ThreeAC.py'
INPUT THE EXPRESSION: a+b*c*d/e
PREFIX: -+ab/*cde

POSTFIX: ab+cd*e/-
t1 := a + b
t2 := c * d
t3 := t2 / e
t4 := t1 - t3

-----Quadruple Table-----

op |      arg1      |      arg2      |      Result
+   |      a         |      b         |      t1
*   |      c         |      d         |      t2
/   |      t2        |      e         |      t3
-   |      t1        |      t3        |      t4

PS C:\Users\jinda\Downloads>
```