

## ΑΣΚΗΣΗ 1η

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί την μετατροπή μίας αριθμητικής έκφρασης από την ένθετη (infix) μορφή στην επιθεματική (postfix), με την χρήση μίας στοίβας. Να γίνει εγκατάσταση της βιβλιοθήκης *pythonds3* ή να χρησιμοποιηθεί η υλοποίηση της στοίβας προηγούμενης άσκησης.

```
from pythonds3.basic.stack import Stack

def infixToPostfix(infixexpr):
    prec = {}
    prec["*"] = 3
    prec["/"] = 3
    prec["+"] = 2
    prec["-"] = 2
    prec["("] = 1
    opStack = Stack()
    postfixList = []
    tokenList = infixexpr.split()

    for token in tokenList:
        if token in "ABCDEFGHIJKLMNOPQRSTUVWXYZ" or token in "0123456789":
            postfixList.append(token)
        elif token == '(':
            opStack.push(token)
        elif token == ')':
            topToken = opStack.pop()
            while topToken != '(':
                postfixList.append(topToken)
                topToken = opStack.pop()
        else:
            while (not opStack.is_empty()) and (prec[opStack.peek()] >=
prec[token]):
                postfixList.append(opStack.pop())
            opStack.push(token)

    while not opStack.is_empty():
        postfixList.append(opStack.pop())

    return " ".join(postfixList)

print(infixToPostfix("( A + B ) * C - ( D - E ) * ( F + G )"))
```

**ΑΣΚΗΣΗ 2η**

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί τον υπολογισμό της επιθεματικής (postfix) μορφής μίας αριθμητικής έκφρασης.

```
from pythonds3.basic.stack import Stack

def postfixEval(postfixExpr):
    operandStack = Stack()
    tokenList = postfixExpr.split()

    for token in tokenList:
        if token in "0123456789":
            operandStack.push(int(token))
        else:
            operand2 = operandStack.pop()
            operand1 = operandStack.pop()
            result = doMath(token, operand1, operand2)
            operandStack.push(result)
    return operandStack.pop()

def doMath(op, op1, op2):
    if op == "*":
        return op1 * op2
    elif op == "/":
        return op1 / op2
    elif op == "+":
        return op1 + op2
    else:
        return op1 - op2

print(postfixEval('2 3 * 3 9 * +'))
```