#�������� ��������� ��� ���������� ������

class TNode:

pass

def newNode( d ):

node = TNode()

node.data = d

node.left = None

node.right = None

return node

#������������� �������� �� �����������

def priority ( op ):

if op in "+-": return 1

if op in "\*/": return 2

return 100

#����� � ��������� ��������� �������� � ���������� �����������

def priority ( op ):

if op in "+-": return 1

if op in "\*/": return 2

return 100

#���������� ������ �� ���������� ���������

def makeTree ( s ):

k = lastOp(s)

if k < 0: # ������� ����

Tree = newNode ( s )

else: # ������� ����-��������

Tree = newNode ( s[k] )

Tree.left = makeTree ( s[:k] )

Tree.right = makeTree ( s[k+1:] )

return Tree

#���������� ��������� �� ������

def calcTree ( Tree ):

if Tree.left == None:

return int(Tree.data)

else:

n1 = calcTree ( Tree.left )

n2 = calcTree ( Tree.right )

if Tree.data == "+": res = n1 + n2

elif Tree.data == "-": res = n1 - n2

elif Tree.data == "\*": res = n1 \* n2

else: res = n1 // n2

return res