hanoi.py

def hanoi(n, a, b, c):

if n > 0:

hanoi(n-1, a, c, b)

print ("move %s: %s --> %s" %(n, a, c))

hanoi(n-1, b, a, c)

hanoi(4, "A", "B", "C")

実行結果

move 1: A --> B

move 2: A --> C

move 1: B --> C

move 3: A --> B

move 1: C --> A

move 2: C --> B

move 1: A --> B

move 4: A --> C

move 1: B --> C

move 2: B --> A

move 1: C --> A

move 3: B --> C

move 1: A --> B

move 2: A --> C

move 1: B --> C

hanoi.java

public class Hanoi{

static void hanoi(int n, char a, char b, char c) {

if(n==0) {

return;

}

else {

hanoi(n-1, a, c, b);

System.out.println("move "+n+": "+a+" --> "+c);

hanoi(n-1, b, a, c);

}

}

public static void main (String arg[]) {

hanoi(4, 'A', 'B', 'C');

}

}

実行結果

move 1: A --> B

move 2: A --> C

move 1: B --> C

move 3: A --> B

move 1: C --> A

move 2: C --> B

move 1: A --> B

move 4: A --> C

move 1: B --> C

move 2: B --> A

move 1: C --> A

move 3: B --> C

move 1: A --> B

move 2: A --> C

move 1: B --> C

rpn.py

def RPN(states):

operator = {

'+': (lambda x, y: x + y),

'-': (lambda x, y: x - y),

'\*': (lambda x, y: x \* y),

'/': (lambda x, y: float(x) / y)

}

stack = []

print('RPN: %s' % states)

for index, z in enumerate(states):

if index > 0:

print(stack)

if z not in operator.keys():

stack.append(int(z))

continue

y = stack.pop()

x = stack.pop()

stack.append(operator[z](x, y))

print('%s %s %s =' % (x, z, y))

print(stack[0])

return stack[0]

def test():

print("OK" if RPN("12+42-\*2/") == 3 else "NG")

test()

実行結果

RPN: 12+42-\*2/

[1]

[1, 2]

1 + 2 =

[3]

[3, 4]

[3, 4, 2]

4 - 2 =

[3, 2]

3 \* 2 =

[6]

[6, 2]

6 / 2 =

3.0

OK