

function findNumber(maxCount)

for i= 2,maxCount do

r = 0

for j = 1 , i - 1 do

if i%j == 0 then

r = r + j

end

end

if r == i then

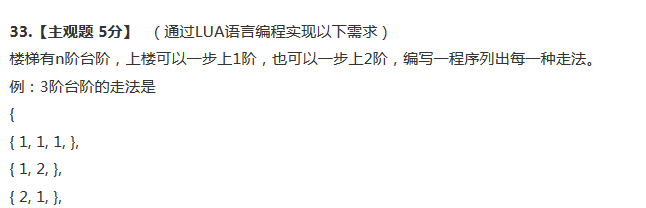
print(i)

end

end

end

findNumber(1000)



function cal(level,nextStep,strS)

level = level - nextStep

if level == 0 then

print(strS .. " " .. nextStep.." }")

elseif level == 1 then

strS = strS.." "..nextStep..","

cal(level,1,strS)

elseif level > 1 then

strS = strS.." "..nextStep..","

cal(level,2,strS)

cal(level,1,strS)

end

end

function test(count)

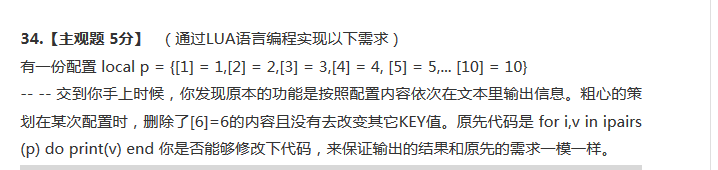
print(count .."阶台阶的走法是：")

cal(count,1,"{")

cal(count,2,"{")

end

test(10)



local p = {[1] = 1,[2] = 2,[3] = 3,[4] = 4,[5] = 5,[7] = 7,[8] = 8,[9] = 9,[10] = 10}

for i,v in ipairs(p) do

print(v)

if i == 5 then

print(6)

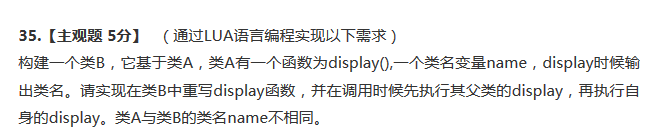
end

if i >= 5 then

p[i+1] = p[i+2]

end

end



A = {}

A.Name = "ClassA"

function A:New(name)

local o = {}

setmetatable(o, self)

self.\_\_index = self

o.Name = name

return o

end

function A:Display()

print(self.Name)

end

B = A:New("ClassB")

function B:Display()

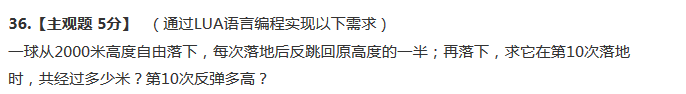
local a = getmetatable(self)

a:Display()

print(self.Name)

end

B:Display()



function cal(height,time)

local tempheight = height

local sum = 0

for i = 1, time do

if i == 1 then

sum = tempheight

else

sum = sum + tempheight \* 2

end

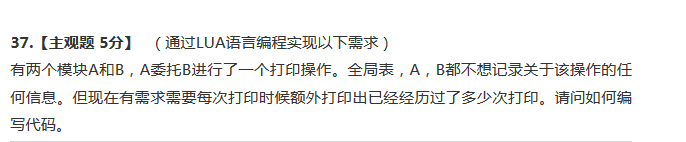
tempheight = tempheight / 2

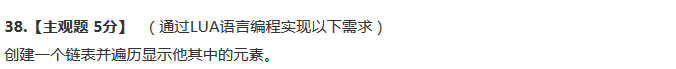
end

print("从"..height.."米落下，到第"..time.."次落下共经过"..sum.."米。第"..time.."次弹起的高度为"..tempheight)

end

cal(2000,10)





Node = {}

function Node:new(val)

local o = {}

setmetatable(o,self)

self.\_\_index = self

o.value = val

o.pnext = nil

return o

end

function Node:addNode(nodeChild)

nodeChild.pnext = self.pnext

self.pnext = nodeChild

return nodeChild

end

List = {}

function List:print()

local list = self.phead

while list do

print(list.value)

list = list.pnext

end

end

List.phead = Node:new(1)

local tempHead = List.phead

for i=2, 10 do

tempHead:addNode(Node:new(i))

tempHead = tempHead.pnext

end

List:print()