Python实验报告5

实验五： 函数

实验目的：掌握函数的定义和调用方法；理解递归函数的使用。

实验要求：

能正确定义和调用函数。

能使用函数解决代码复用。

能编写递归函数。

实验过程：

程序练习题5.1 程序练习题3.5输出了一个简单的田字格，用函数简化其代码输出如图5.12所示更大的田字格。

编写代码：

def tbedge(n):

print('+ - - - - ' \* n + '+')

def lredge(n):

print('| ' \* n + '|')

def matts(n):

for i in range(5\*n+1):

if i%5 == 0:

tbedge(n)

else:

lredge(n)

matts(4)

程序练习题5.2 实现isOdd()函数，参数为整数，如果整数为奇数，返回True，否则返回Fasle。

编写代码：

def isodd(n):

if n%2 == 1:

return True

else:

return False

def main():

while True:

s = input("请输入一个整数：")

if not s.isdigit():

print("请再次输入一个整数：")

continue

n=eval(s)

if isodd(n):

print("{} is odd".format(n))

else:

print("{} is even".format(n))

main()

程序练习题5.3 实现isNum()函数，参数为一个字符串，如果这个字符串属于整数、浮点数或复数的表示，则返回True，否则返回False。

编写代码：

def isNum(s):

try:

n = eval(s)

except:

return False

return True

s = input("Enter a string:")

if isNum(s):

print("{} is a number".format(s))

else:

print("{} is not a number".format(s))

程序练习题5.4实现multi()函数，参数个数不限，返回所有参数的乘积。

编写代码：

def multi(\*a):

if len(a) == 0:

return 0

t = 1

for i in a:

t = t \* i

return t

print( multi(2,3))

程序练习题5.5 实现isPrime()函数，参数为整数，要有异常处理。如果整数是质数，返回True,否则返回Fasle。

编写代码：

def isPrime(n):

for i in range(2,6):

if n%i == 0:

return False

return True

# 输入-1程序结束

while True:

try:

n = eval(input("Enter an Integer: "))

except:

print("Enter error! Please enter again: ")

continue

if type(n) is not int:

print("Please enter an Integer: ")

continue

if n == -1:

break

if isPrime(n):

print("{} is prime.".format(n))

else:

print("{} is not prime.".format(n))

def isprime(n):

for i in range(2,n):

if n%i == 0:

return False

else:

return True

while True:

try:

n = eval(input("请输入一个整数: "))

except:

print("请输入一个整数！请再次输入：")

continue

if n == -1:

break #结束循环

if isprime(n):

print("{}是一个质数。".format(n))

else:

print("{}不是一个质数。".format(n))

斐波拉契数列。

Fabric数列描述如下：

编写Fabric函数程序，用递归实现。

编写代码：

def Fabric(n):

if n in [1,2]: #或者if n == (1,2)

return 1

#elif n == 2:

# return 1

else:

return Fabric(n-1) + Fabric(n-2)

num = eval(input(""))

print(Fabric(abs(int(num))))

程序练习题5.7 汉诺塔。

编写代码：

# n个圆盘由a柱移动到c柱，借助b柱

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

if n == 1:

print(a + "-->" + c)

else:

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

hanoi(a,b,c,1)

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

hanoi('a','b','c',6)

七段数码管问题。

在教材实例7的基础上改写代码，实现以下要求：

1）日期的每一个数字用不同颜色表示。

提示：0-9共有10个数字，对应10种不同颜色表示：'red', 'blue', 'yellow', 'gold', 'violet', 'purple', 'green', 'darkgreen', 'grey', 'orange'；只需要修改drawDigit()函数，添加功能使之实现每一个数字用不同颜色表示。

2）数码管的每一段用不同颜色表示。

提示：方法一：修改drawDigit()函数；方法二：修改drawLine()函数。

修改实例代码18.1，使koch曲线反向绘制，从直线开始，中间部分向下方绘制。

提示：参考教材实例8

编写代码：

import turtle, datetime, random

def drawGap(): #绘制数码管间隔

turtle.penup()

turtle.fd(5)

def drawLine(draw): #绘制单段数码管

drawGap()

r = random.randint(0,255)

g = random.randint(0,255)

b = random.randint(0,255)

turtle.pencolor((r,g,b))

turtle.pendown() if draw else turtle.penup()

turtle.fd(40)

drawGap()

turtle.right(90)

def drawStr(str):

r = random.randint(0,255)

g = random.randint(0,255)

b = random.randint(0,255)

turtle.pencolor((r,g,b))

turtle.write(str,font=("Arial", 18, "normal"))

turtle.fd(40)

def drawDigit(digit): #根据数字绘制七段数码管

drawLine(True) if digit in [2,3,4,5,6,8,9] else drawLine(False)

drawLine(True) if digit in [0,1,3,4,5,6,7,8,9] else drawLine(False)

drawLine(True) if digit in [0,2,3,5,6,8,9] else drawLine(False)

drawLine(True) if digit in [0,2,6,8] else drawLine(False)

turtle.left(90)

drawLine(True) if digit in [0,4,5,6,8,9] else drawLine(False)

drawLine(True) if digit in [0,2,3,5,6,7,8,9] else drawLine(False)

drawLine(True) if digit in [0,1,2,3,4,7,8,9] else drawLine(False)

turtle.left(180)

turtle.penup()

turtle.fd(20)

def drawDate(date): #获得要输出的数字

for i in date:

if i == '-':

drawStr('年')

elif i == '=':

drawStr('月')

elif i == '+':

drawStr('日')

else:

drawDigit(eval(i))

def main():

turtle.setup(800, 350, 200, 200)

turtle.colormode(255)

turtle.penup()

turtle.fd(-300)

turtle.pensize(5)

drawDate(datetime.datetime.now().strftime('%Y-%m=%d+'))

turtle.done()

main()

实验思考：

函数是一段具有特定功能的、可重用的语句组，用函数名来表示并通过函数名进行功能调用。函数在python编程中有着重要的作用，主要体现在降低编程难度和代码重用上。对python函数的学习，首先得理解形式参数和实际参数。

程序调用一个函数需要执行以下4个步骤

（1）调用程序在调用处暂停执行。

（2）在调用时将实参复制给函数的形参。

（3）执行函数体语句。

（4）函数调用结束给出返回值，程序回到调用前的暂停处继续执行。