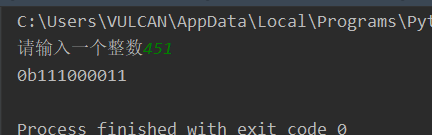
**实验五 函数**

1. **编写一个函数func（n），接受一个十进制数N作为参数，返回二进制数。**  
   def transform(dec):  
    a = bin(dec)  
    return a  
   if \_\_name\_\_ == '\_\_main\_\_':  
    a = int(input("请输入一个整数"))  
    print(transform(a))



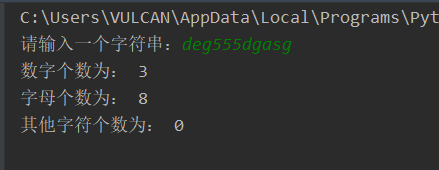
1. **编写2个函数，其中一个函数计算并返回斐波那契数列第i项（使用递归），另一个函数计算并返回斐波那契数列前10项和，并对其进行测试。**

def Fibonacci(n):  
 if n <=0:  
 return 0  
 elif n==1:  
 return 1  
 return Fibonacci(n - 1) + Fibonacci(n - 2)  
def fun2():  
 sum=0  
 for i in range(1,11):  
 #print(Fibonacci(i))  
 sum+=Fibonacci(i)  
 return sum  
  
if \_\_name\_\_=="\_\_main\_\_":  
 i=int(input("请输入你想得到的项数："))  
 b=Fibonacci(i)  
 print("第%d项为%d"%(i,b))  
 sum=fun2()  
 print("10项和为：",sum)



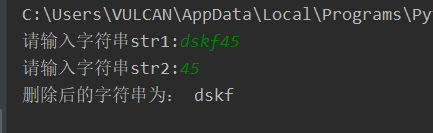
1. **编写一个函数func（str），计算并返回字符串str中的数字、字母及其其他类型字符的个数。**

def func(str):  
 count\_num=0  
 count\_alp=0  
 count\_oth=0  
 for i in str:  
 if i .isdigit():  
 count\_num+=1  
 elif i.isalpha():  
 count\_alp +=1  
 else:  
 count\_oth+=1  
 print("数字个数为：",count\_num)  
 print("字母个数为：",count\_alp)  
 print("其他字符个数为：",count\_oth)  
  
if \_\_name\_\_=="\_\_main\_\_":  
 str=str(input("请输入一个字符串："))  
 func(str)



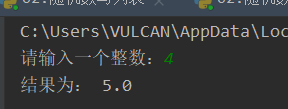
1. **编写一个函数func（str1，str2），将字符串str1中出现的字符串str2删除，然后作为函数的结果返回。**

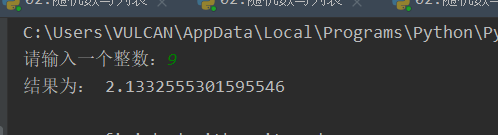
def delete():  
 str1 = input("请输入字符串str1:")  
 str2 = input("请输入字符串str2:")  
 if str2 in str1:  
 str=str1.replace(str2,'')  
 print("删除后的字符串为：",str)  
  
if \_\_name\_\_=="\_\_main\_\_":  
 delete()



1. **编写一个函数func（n），实现如下功能：**
2. **传入参数n为偶数，函数返回的结果**
3. **传入参数n为奇数，函数返回的结果**

def func(n):  
 sum = 0  
 if n%2 == 0:  
 for i in range(1,n+1):  
 sum += 1/2\*i  
 else:  
 for i in range(0,n+1):  
 sum += 1/(2\*i+1)  
 print("结果为：", sum)  
   
   
if \_\_name\_\_=="\_\_main\_\_":  
 n=int(input("请输入一个整数："))  
 func(n)





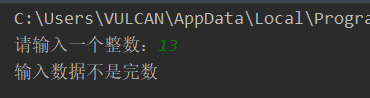
1. **有5个人坐在一起，向第5人询问其岁数，他说他比第4个人大2岁，询问第四个人，他说他比第3个人大2岁，询问第3个人，他说他比第2个人大2岁，询问第2个人，他说他比第1个人大2岁，询问第1个人，他说他10岁，问第5个人岁数是多少？分别使用非递归和递归实现。**

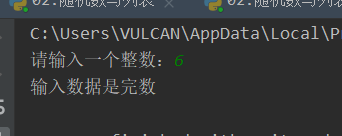
def norecur():  
 age=10  
 for i in range(2,6):  
 age+=2  
 print("第5个人岁数是：",age)  
def recur(i):  
 if i==1:  
 return 10  
 else:  
 return recur(i-1)+2  
if \_\_name\_\_=="\_\_main\_\_":  
 print("非递归：")  
 norecur()  
 age=recur(5)  
 print("递归：")  
 print("第5个人岁数是：", age)



1. **如果一个正整数的所有因子（包括1，不包括它本身）之和与该数相等，则称为完数（如6=1+2+3），编写一个函数isWs（n）判断传入的是否为完数，返回true为是，False为否，编程实现。**

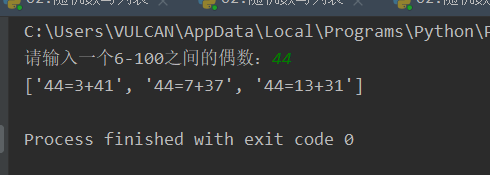
def isWs(n):  
 sum=0  
 for i in range(1,n):  
 if n%i==0:  
 sum+=i  
 if sum==n:  
 return True  
 else:  
 return False  
if \_\_name\_\_=="\_\_main\_\_":  
 n = int(input("请输入一个整数："))  
 flag=isWs(n)  
 if flag==True:  
 print("输入数据是完数")  
 if flag==False:  
 print("输入数据不是完数")





1. **验证哥德巴赫猜想：任意一个大于2的偶数都可以表示成2个素数之和。编写一个函数isGDBH（n）将传入的6-100之间的偶数表示为2个素数之和，结果保存在列表中返回。如传入10，则返[“10=3+7,10=5+5”]。**

import math  
def isprime(n): # 判断素数  
 if n == 1:  
 return False  
 elif n == 2:  
 return True  
 else:  
 for i in range(2, int(math.sqrt(n) + 1)):  
 if n % i == 0:  
 return False  
 return True  
def thonsand(n): # 生成若干个素数，返回素数list  
 a = []  
 for i in range(1, n + 1):  
 if isprime(i):  
 a.append(i)  
 return a  
def isGDBH(n):  
 a = []  
 ls = thonsand(n)  
 for i in ls:  
 for j in ls:  
 if i<=j:  
 if n == i + j:  
 a.append(str(n)+"="+str(i)+"+"+str(j))  
 return a  
if \_\_name\_\_=="\_\_main\_\_":  
 n = int(input("请输入一个6-100之间的偶数："))  
 print(isGDBH(n))



1. **编程实现如下功能：**
2. **编写3个函数，分别求三角形、矩形和圆形周长。**
3. **使用装饰器对传入参数进行调用和合法性检查。**

import functools  
#装饰器  
def log(func):  
 @functools.wraps(func)  
 def wrapper(\*args, \*\*kwargs):  
 print('call %s():' % func.\_\_name\_\_)  
 print('args = {}'.format(\*args))  
 return func(\*args, \*\*kwargs)  
 return wrapper  
@log  
def tri(a,b,c):  
 girth=a+b+c  
 print("三角形的周长为：",girth)  
  
@log  
def rec(a,b):  
 girth = (a + b)\*2  
 print("矩形的周长为：", girth)  
  
@log  
def cyc(r):  
 girth=2\*3.14\*r  
 print("圆形的周长为：", girth)  
  
if \_\_name\_\_=="\_\_main\_\_":  
 tri(3,4,5)  
 print("")  
 rec(6,7)  
 print("")  
 cyc(8)

