**#牛顿法计算n的平方根**

def t(n):

x = n/2

while True:

y = (x+n/x)/2

if abs(y-x)<0.00000001:

break

x=y

return x

print(t(25))

**#画弧**

import turtle

import math

def arc(t,r,angle):

arc\_length = 2\*math.pi\*r\*angle/360

n = int(arc\_length/3)+1

step\_length = arc\_length/n

step\_angle = float(angle)/n

for o in range(n):

t.fd(step\_length)

t.lt(step\_angle)

t = turtle.Turtle()

arc(t,100,60)

**#编写圆函数并调用半径是90的函数**

import turtle

import math

b=turtle.Turtle()

def t(b,n,length):

angle = 360/n

for i in range(n):

b.fd(length)

b.lt(angle)

def circle(b,r):

arc\_length=2\*math.pi\*r

n = int(arc\_length/3)+1

g = arc\_length/n

t(b,n,g)

circle(b,90)

**#反转字典**

def invert\_dict(d):

invert = dict()

for key in d:

val = d[key]

if val not in invert:

invert[val]=[key]

else:

invert[val].append(key)

return invert

d = {"e":1,"r":2,"t":"u"}

print(invert\_dict(d))

**#打印几位数 逆序输出(长度不超过5)**

def count(n):

list = []

while(n>0):

list.append(n%10)

n = n // 10

print(len(list))

print(list)

count(12345)

**#列表元素求和并放在列表尾**

def llist(d):

number = 0

for i in d:

number += i

d.append(number)

return d

h = [1,2,3]

print(llist(h))

**#类的初始化**

class Time():

def \_\_init\_\_(self,hour=0,minute=0,second=0):

self.hour=hour

self.minute=minute

self.second=second

def \_\_str\_\_(self):

return '%.2d:%.2d:%.2d' % (self.hour,self.minute,self.second)

time = Time(11,45,30)

print(time)

**#读取当前文件 输出长度超过10的单词**

def count\_len(filename):

list=[]

file=open(filename,"r")

for line in file:

words=line.split()

for word in words:

if len(word)>10:

print(word)

count\_len("C:/Users/17810/Desktop/abc.txt")

**#计算斐波那契数列**

def f(n):

if n==1:

return 1

if n==2:

return 1

else:

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

def fibo(number):

for i in range(1,number+1):

print(f(i))

fibo(10)

**#统计序列中各元素出现的个数**

def count\_list(list):

d = dict()

for i in list:

if i not in d:

d[i]=1

else:

d[i]=d[i]+1

return d

print(count\_list([1,21,1,21,5,45,"a","f","a","g","g"]))

**#搜索索引**

def find(words,key):

index=0

while index < len(words):

if words[index]==key:

return index

else:

index += 1

return 0

print(find("asdfghjkl","h"))