**Introduction to Computer Science**

**Homework 8**

**沙行勉教授**

**Due Date: 星期三，Dec. 5, 2018**

**Do not copy others, or you will get punished. If you allow other students to copy your report, you will be punished as well. In the homework, you will get more experience in programming. Good for you. 但是，对于编程题，我希望有基础的人能多帮助其他同学，帮助他们debug, 但是不要让他们抄袭。好吗？**

**PUT DOWN YOUR NAME and EMAIL ADDRESS IN YOUR REPORT.** tThe Filename of your report should be HOMEWORK N\_Student ID\_your name\_your \_Teacher’s last name (Sha or Zhang), where N is the homework number, for example, *1\_2018XXXX \_XiaoMing\_Sha*. Then send your report to the TA of your class. 注意，两班的TA不一样，千万不要送错了。**你可以用中文来回答问题。**

**请同学阅读书本的第四章。关于小乌龟画图，同学可以看编程导论第七章。**

1. **（每题5 points）Let ID=your student ID. Then, let k=ID %2. So k can be 0 or 1. Based on the value of your ID, you get value of k. Please tell us your k value.**

**If k==0: 做习题 4.2.4，4.2.6， 4.2.8， 4.2.10， 4.2.12， 4.2.14， 4.5.2, 4.5.4**

**If k==1: 做习题 4.2.5， 4.2.7， 4.2.9， 4.2.11， 4.2.13， 4.2.15, 4.5.3, 4.5.5**

**My k is 1**

**4.2.5: L.reverse是反转列表本身，不会产生新的列表，也不会有返回值，而L[-1:-1-len(L):-1]是反向遍历一遍L中的元素，并依次添加到新的列表中，L本身并没有改变**

**4.2.7 令i=len(L)**

**4.2.9**

**def find(s,x):**

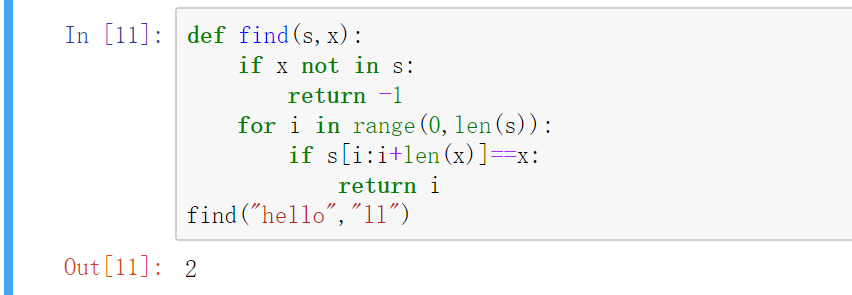
**if x not in s:**

**return -1**

**for i in range(0,len(s)):**

**if s[i:i+len(x)]==x:**

**return i**

**find("hello","ll")**

**4.2.11**

**def find(s,x):**

**if x not in s:**

**return -1**

**for i in range(0,len(s)):**

**if s[i:i+len(x)]==x:**

**return i**

**def count(s,x):**

**if x not in s:**

**return 0**

**i=0**

**while x in s:**

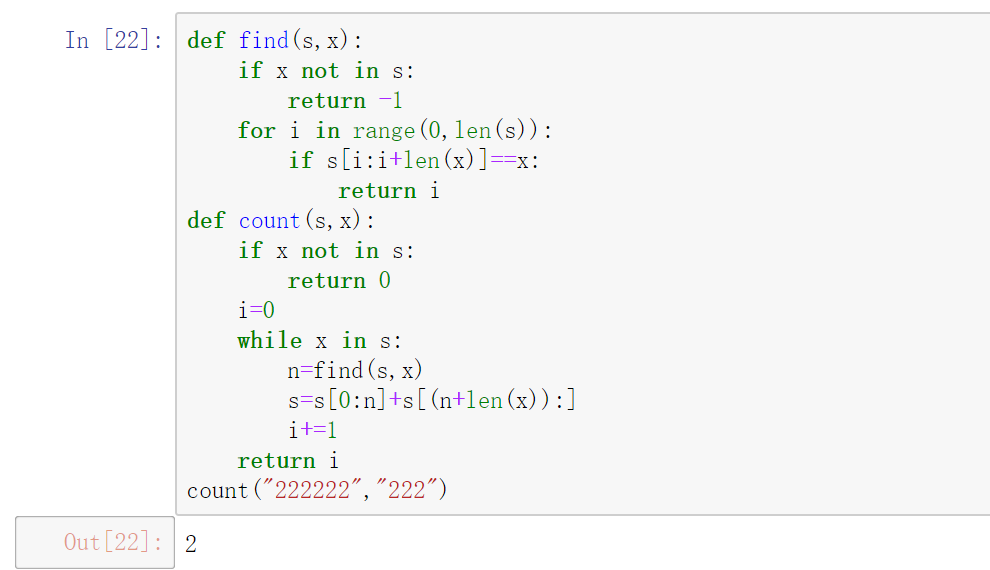
**n=find(s,x)**

**s=s[0:n]+s[(n+len(x)):]**

**i+=1**

**return i**

**count("222222","222")**



**4.2.13**

**程序一：['stu', '606866']**

**606866**

**程序二：{'role': 'stu', 'phone': '606866'}**

**606866**

**4.2.15**

**d\_num\_year={"Aaron":["1","2012"],"Abraham":["2","2014"], "Andy":["3","2013"],"Benson":["4","2014"]}**

**4.5.3**

**def recursiveSum(L):**

**L1=L[:]**

**if len(L1)==0:**

**return 0**

**cur=L1.pop()**

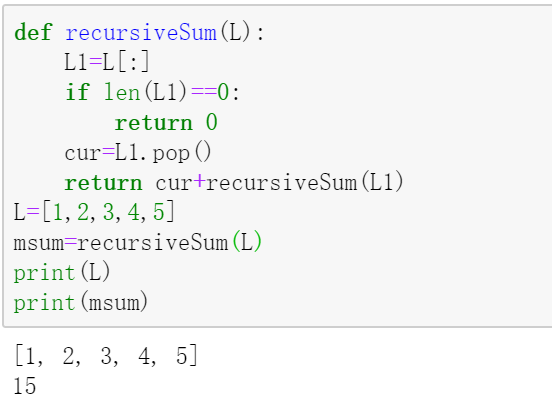
**return cur+recursiveSum(L1)**

**L=[1,2,3,4,5]**

**msum=recursiveSum(L)**

**print(L)**

**print(msum)**



**4.5.5**

**[1,2,3]，因为根据局部变量的定义，函数内等号左边的量为局部变量，因此，L=L+[4]，这个式子中，左边的L为局部变量，而右边的L为全局变量,因此，函数改变的量是局部变量而非全局变量，L本身仍为[1,2,3]**

**2.、（30 points）This programming assignment can be done by you alone or a group of two students. 写下合作同学的名字。程序练习题4.8.1, 4.8.2, 4.8.3**

**4.8.1**

**from turtle import \***

**p=Turtle()**

**p.pensize(5)**

**p.pencolor("black")**

**p.setheading(0)**

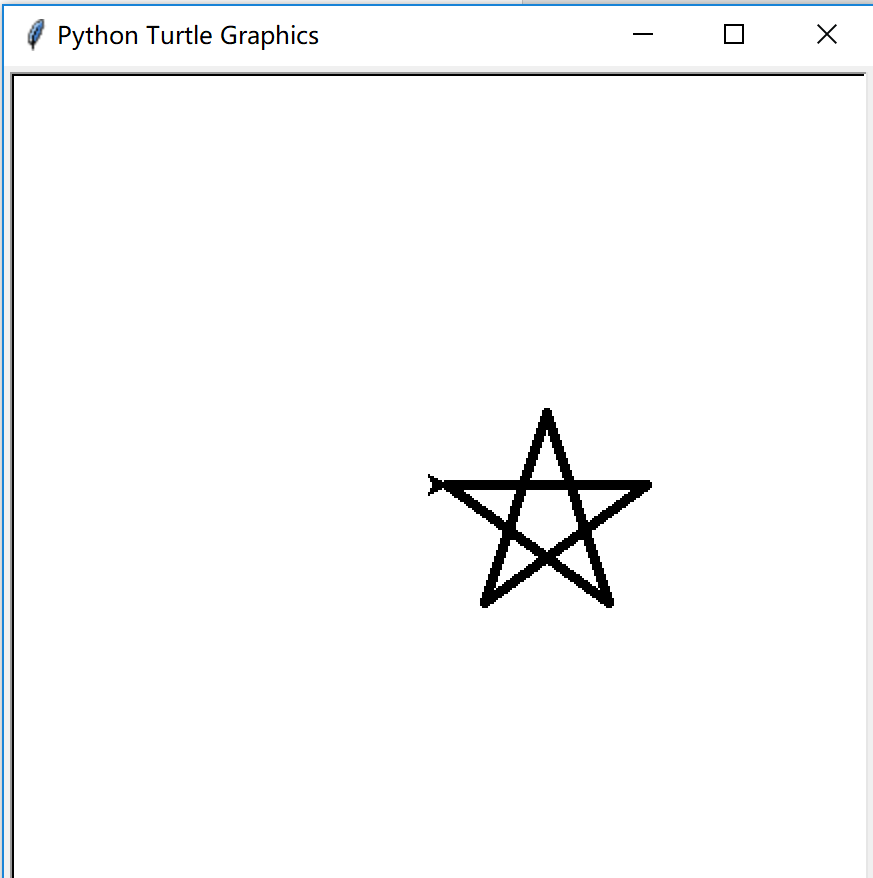
**length=100;angle=0**

**for i in range (5):**

**p.forward(length)**

**angle-=144**

**p.setheading(angle)**



**4.8.2**

**from turtle import \***

**def jumpto(x,y):**

**up();goto(x,y);down()**

**p=Turtle()**

**p.home()**

**p.pensize(1)**

**p.pencolor("red")**

**speed("fast")**

**i=1**

**for r in range(50,111,20):**

**color("red")**

**circle(r)**

**jumpto(0,-20\*i)**

**i+=1**

**jumpto(0,50)**

**def fang():**

**color("blue")**

**for i in range(4):**

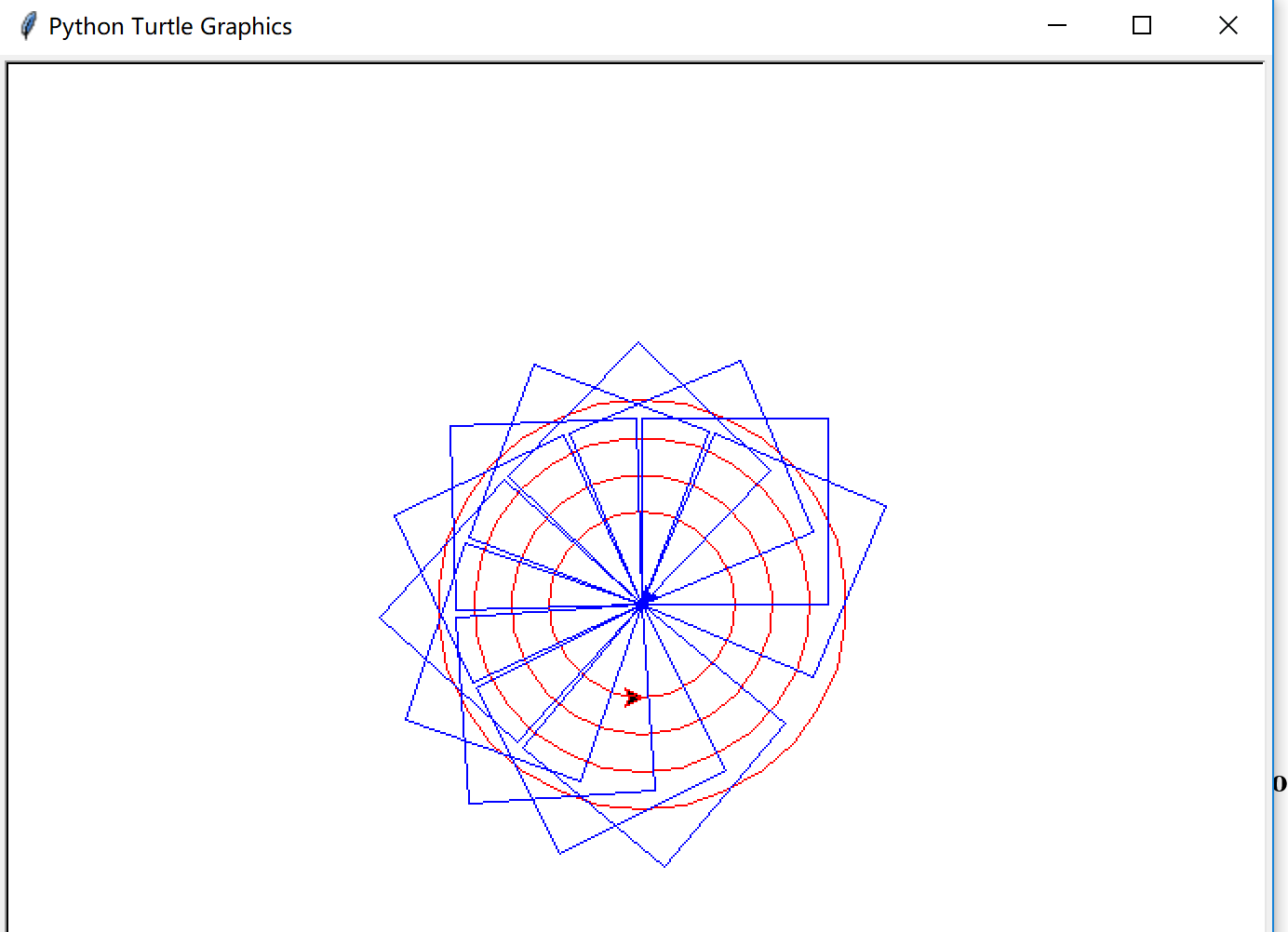
**forward(100)**

**left(90)**

**for e in range(-23,240,23):**

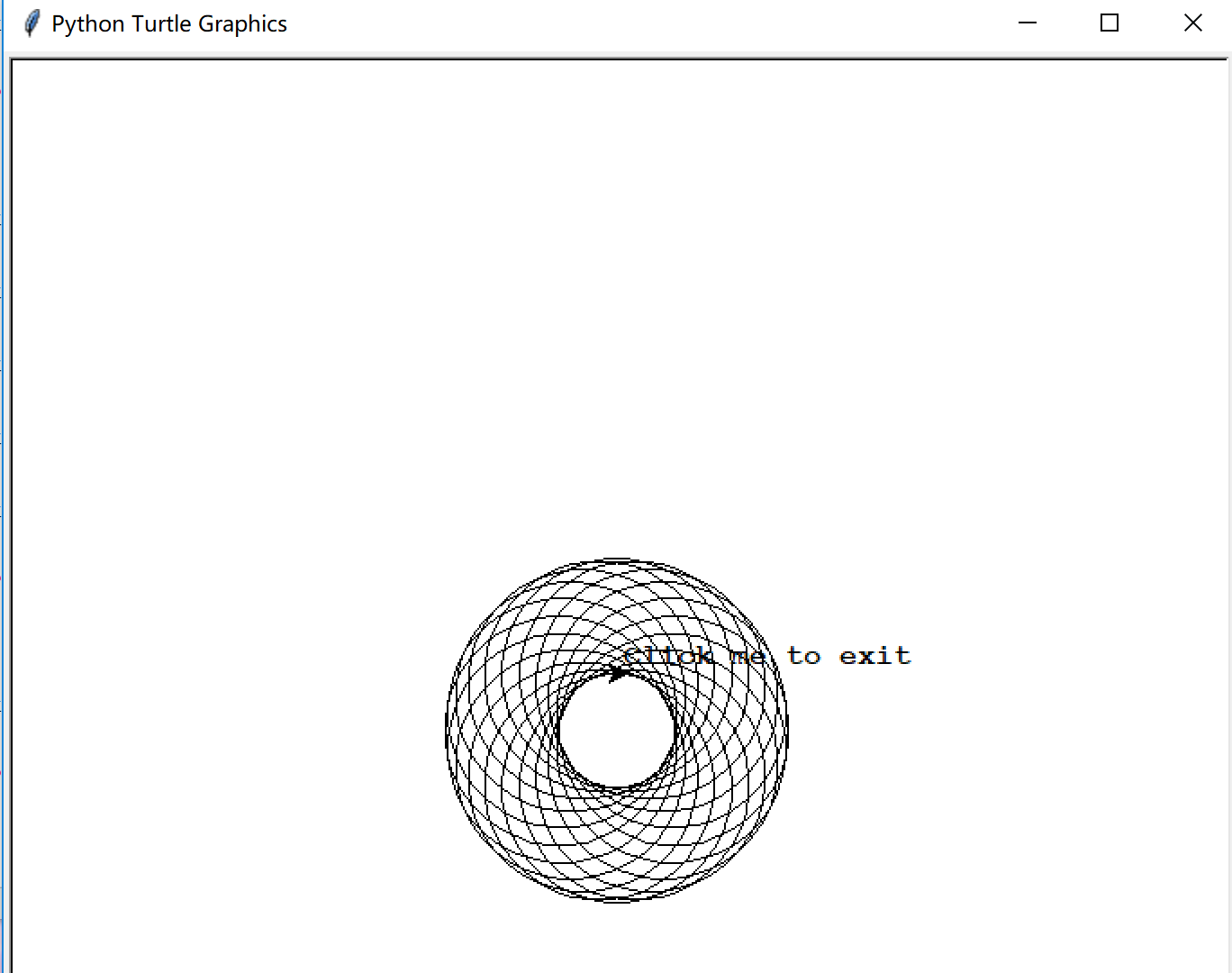
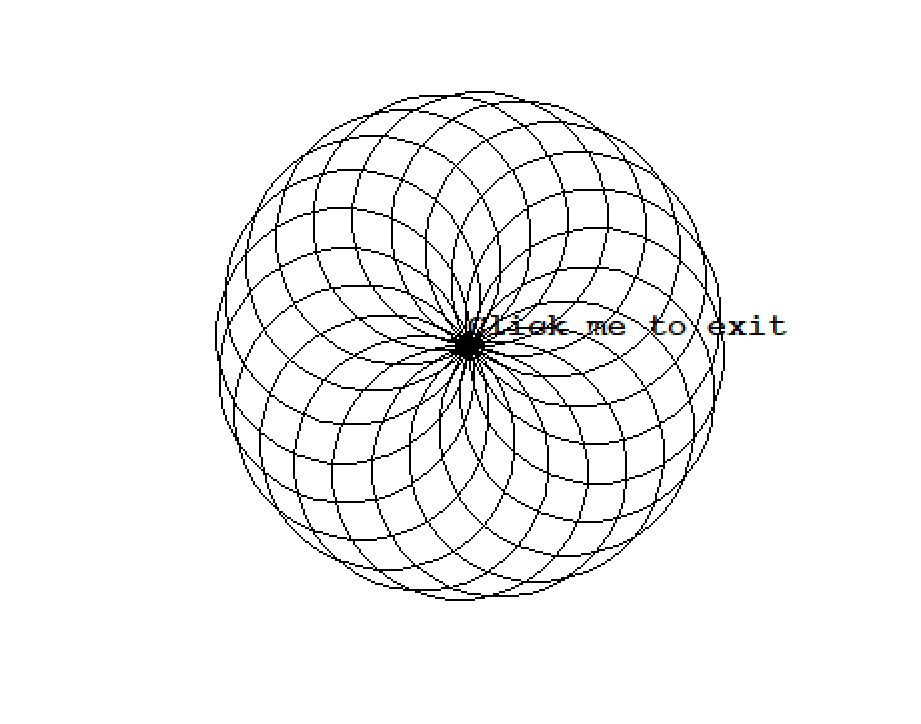
**setheading(e)**

**fang()**



**4.8.3**

**IN\_TIMES是作出正40边形来拟合一个近似圆，TIMES是在每画完一个圆之后，调整圆心的位置再画下一个圆，使得所有圆的圆心在一个新的小圆上，如果增大INTIMES，圆的拟合程度会越大，正多边形会更接近圆，反之与圆的相似程度更小，如果增大TIMES，那么图形的重叠密度会更高，反之图形重叠密度更稀疏，如果注释掉forward(200/IN\_TIMES),那么圆心的位置不会改变，将会变成多个圆心重叠的聚合图案**



变成这样—>

**3、（30 points）This programming assignment can be done by you alone or a group of two students. 写下合作同学的名字。习题4.3， 4.6**

**4.3：**

**（1）：输出为[0,3],[0,0]，因为在这个循环中采用的是X=L的复制方式，L与X指向同一个地址，因此X一直在改变，再遍历完第一堆牌之后，没有把第一堆的牌还原就进行了第二次遍历，而且这个算法是将一堆牌全部拿完后再输出，并不是随机数**

**（2）def findP(L):**

**for i in range (0,(len(L))):**

**a=L[i];X=L[:]**

**if a==0:**

**continue**

**while a>0:**

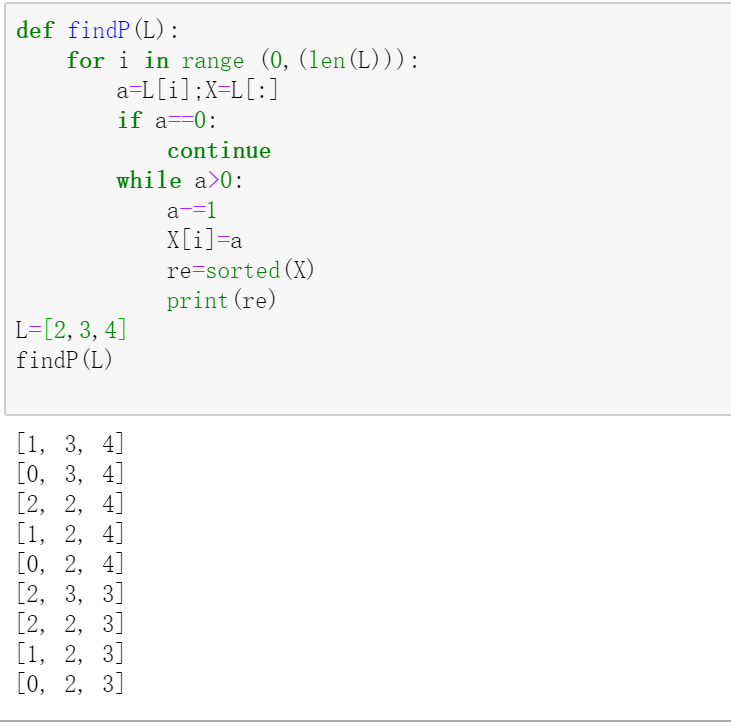
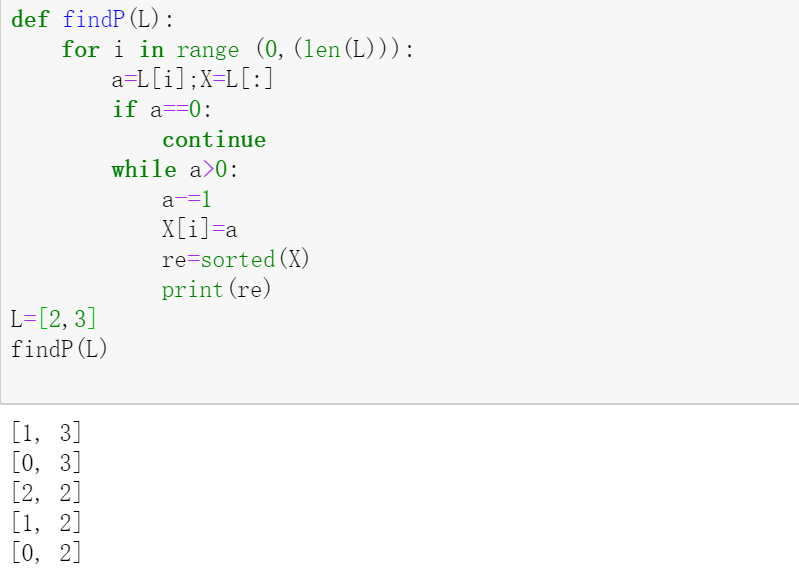
**a-=1**

**X[i]=a**

**re=sorted(X)**

**print(re)**

**L=[2,3]**

**findP(L)**

**4.6**

**L1=["Aaron","Benson","Howard","Ophelia"]**

**L2=[["Aaron",300,3,"Benson","Howard","Ophelia"],["Benson",150,2,"Aaron","Ophelia"],["Howard",100,1,"Benson"],["Ophelia",200,2,"Aaron","Howard"]]**

**for e in L1:**

**n=0**

**for p in L2:**

**if p[0]==e:**

**n-=p[1]**

**for m in L2:**

**i=1**

**while i<=m[2]:**

**if m[i+2]==e:**

**n+=m[1]/m[2]**

**i+=1**

**print(e,end=" ")**

**print(n,end=" ")**