1、

a = np.array([1,2,3,4,5])

b = a[2:4]

b[0] = 300

a[1] = 200

print(a) **# a = array[1,2,200,4,5]**

print(b)  **# b = array[300,4]**

2、请在下面的空白处填写运行结果

seq = [1,2,3,4]

seq[-2:]

\_\_**\_[3,4]\_**\_\_

seq[10:]

\_\_**\_\_\_[ ]\_**\_\_\_\_

seq[::-1]

\_**\_\_\_\_[4,3,2,1]\_\_**\_\_\_\_\_

Id(seq[:]) == id(seq)

\_\_\_\_\_\_**\_\_False\_**\_\_\_

1. 一行代码实现对列表a中的偶数位置的元素进行加3后求和？

**sum[np.array(List[1::2])+3]**

4、补充缺失的代码

def print\_directory\_contents(Path):

’’’’’’

这个函数接受文件夹的名称作为输入参数

返回该文件夹中文件的路径

以及其包含文件夹中文件的路径。

def print\_directory\_contents(sPath):

import os

for sChild in os.listdir(sPath):

sChildPath = os.path.join(sPath,sChild)

if os.path.isdir(sChildPath):

print\_directory\_contents(sChildPath)

else:

Print( sChildPath)

5、以下的代码的输出将是什么？

class Parent(object):

x=1

class Child1(Parent):

x=2

class Child2(Parent):

pass

print(Parent.x,Child1.x,Child2.x) # **1 2 1**

Child1.x=4

print(Parent.x,Child1.x,Child2.x) # **1 4 1**

Parent.x=6

print(Parent.x,Child1.x,Child2.x) # **6 4 6**

1. 实现一个二分法查找list元素的程序

二分法是一种快速查找的方法，时间复杂度低，逻辑简单易懂，总的来说就是不断的除以2除以2...

例如需要查找有序list里面的某个关键字key的位置，那么首先确认list的中位数mid，下面分为三种情况：

如果 list[mid] < key,说明key 在中位数的 右边；

如果 list[mid] > key,说明key 在中位数的 左边；

如果 list[mid] = key,说明key 在中位数的中间；

范围每次缩小一半，写个while的死循环知道找到为止。

*二分法查找非常快且非常常用，但是唯一要求是要求数组是有序的*

*代码如下*

[复制代码](javascript:void(0);)

1 #!/usr/bin/python2.7

2 # -\*- coding: utf-8 -\*-

3

4 def BinarySearch(lista, key):

5 # 记录数组的最高位和最低位

6 min = 0

7 max = len(lista) - 1

8

9 if key in lista:

10 # 建立一个死循环，直到找到key

11 while True:

12 # 得到中位数

13 mid = (min + max) / 2

14 # key在数组左边

15 if lista[mid] > key:

16 max = mid - 1

17 # key在数组右边

18 elif lista[mid] < key:

19 min = mid + 1

20 # key在数组中间

21 elif lista[mid] == key:

22 print str(key) + "在数组里面的第" + str(mid) + "个位置"

23 return lista[mid]

24 else:

25 print("没有该数字!")

26

27

28 if \_\_name\_\_ == "\_\_main\_\_":

29 arr = [1, 6, 9, 15, 26, 38, 49, 57, 63, 77, 81, 93]

30 while True:

31 key = input("请输入你要查找的数字：")

32 if key == " ":

33 print("谢谢使用！")

34 break

35 else:

36 BinarySearch(lista, int(key))

7、一本书有100页，每天可以看1页也可以看2页，看完这本书有多少种看法