day2作业及答案

1.练习Pycharm的单步调试(无需提交，练习即可） 2.自己定义变量赋值不同的数据类型并打印，并使用type（与上课一致） 3.能够将整型转为不同进制，进行输出（与上课一致） 4.实现从1到100之间的奇数求和 5.打印九九乘法表（直接百度乘法表图像，与其一致即可，和课件一致也可以） 难度作业： 6.统计一个整数对应的二进制数的1的个数。输入一个整数（可正可负，负数就按64位去遍历即可）， 输出该整数的二进制包含1的个数（使用位运算，不会位运算的同学可以暂时不做该题）

4,5,6已提供答案，提交作业后，即可看到答案

```
4 实现从1到100之间的奇数求和
print(sum([x for x in range(1,101) if x%2]))

5 打印九九乘法表
for i in range(1, 10):
    for j in range(1, i + 1):
        print("%d * %d = %-2d" % (j, i, i * j), end='  ')
    print()

6 统计一个整数对应的二进制数的1的个数。输入一个整数（可正可负）， 输出该整数的二进制包含1的个数
s=int(input("输入整数"))
bin_s=bin(s)

if s>=0:
    #正数补码=正数原码
    num=bin_s.count('1')
else:
    #对于负数

    num=64-bin(-s-1).count('1')    #64位，-s-1为原负数补码取反

print("%d 的2进制中1的个数为%d" %(s,num))

6题 解法2
def count_one_times(num):
    flag = 1
    i = 1
    total = 0  # 统计1的个数
    while i <= 64:
```

```
        if flag & num:  # 按位与为真
            total += 1
        flag <<= 1  # flag左移
        i += 1
    return total


print(count_one_times(-5))
```

day3 作业

1、完成列表、字典的增删查改（代码与上课保持一致）

完成字符串的切片

num_str = "0123456789"

截取从 2 ~ 5 位置 的字符串

截取从 2 ~ 末尾的字符串

截取从 开始~ 5 位置 的字符串

截取完整的字符串

从开始位置，每隔一个字符截取字符串

从索引 1 开始，每隔一个取一个

截取从 2 ~ 末尾 - 1的字符串

截取字符串末尾两个字符

字符串的逆序（面试题）

使用enumerate把 seasons = ['Spring', 'Summer', 'Fall', 'Winter'] 变为一个字典，效果和上课一致

下面是难度作业（完成基础作业即可提交，难度作业不会写可以直接看答案）

```
2、求两个有序数字列表的公共元素
3、给定一个n个整型元素的列表a，其中有一个元素出现次数超过n / 2，求这个元素
4、列表、元组，字典的相同点，不同点有哪些，请罗列
5、将元组 (1,2,3) 和集合 {4,5,6} 合并成一个列表。
6、在列表 [1,2,3,4,5,6] 首尾分别添加整型元素 7 和 0。
7、反转列表 [0,1,2,3,4,5,6,7] 。
8、反转列表 [0,1,2,3,4,5,6,7] 后给出中元素 5 的索引号。
9、分别统计列表 [True,False,0,1,2] 中 True,False,0,1,2的元素个数，发现了什么？
10、从列表 [True,1,0,'x',None,'x',False,2,True] 中删除元素'x'。
11、从列表 [True,1,0,'x',None,'x',False,2,True] 中删除索引号为4的元素。
```

12、删除列表中索引号为奇数（或偶数）的元素。
13、清空列表中的所有元素。
14、对列表 [3,0,8,5,7] 分别做升序和降序排列。
15、将列表 [3,0,8,5,7] 中大于 5 元素置为1，其余元素置为0。
16、遍历列表 ['x','y','z']，打印每一个元素及其对应的索引号。
17、将列表 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] 拆分为奇数组和偶数组两个列表。
18、分别根据每一行的首元素和尾元素大小对二维列表 [[6, 5], [3, 7], [2, 8]] 排序。相当于按6,3,2进行排序，除非第一个元素相等，按第二个元素排序。
19、从列表 [1,4,7,2,5,8] 索引为3的位置开始，依次插入列表 ['x','y','z'] 的所有元素。
20、快速生成由 [5,50) 区间内的整数组成的列表。
21、若 a = [1,2,3]，令 b = a，执行 b[0] = 9， a[0]亦被改变。为何？如何避免？----讲了深COPY和浅COPY再做
22、将列表 ['x','y','z'] 和 [1,2,3] 转成 [('x',1),('y',2),('z',3)] 的形式。
23、以列表形式返回字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中所有的键。
24、以列表形式返回字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中所有的值。
25、以列表形式返回字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中所有键值对组成的元组。
26、向字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中追加 'David':19 键值对，更新Cecil的值为17。
27、删除字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中的Beth键后，清空该字典。
28、判断 David 和 Alice 是否在字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中。
29、遍历字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21}，打印键值对。
30、若 a = dict()，令 b = a，执行 b.update({'x':1})， a亦被改变。为何？如何避免？----讲了深COPY和浅COPY再做（现在别做，容易出错）
31、以列表 ['A','B','C','D','E','F','G','H'] 中的每一个元素为键，默认值都是0，创建一个字典。
32、将二维结构 [['a',1],['b',2]] 和 (('x',3),('y',4)) 转成字典。
33、将元组 (1,2) 和 (3,4) 合并成一个元组。
34、将空间坐标元组 (1,2,3) 的三个元素解包对应到变量 x,y,z。
35、返回元组 ('Alice','Beth','Cecil') 中 'Cecil' 元素的索引号。
36、返回元组 (2,5,3,2,4) 中元素 2 的个数。
37、判断 'Cecil' 是否在元组 ('Alice','Beth','Cecil') 中。
38、返回在元组 (2,5,3,7) 索引号为2的位置插入元素 9 之后的新元组。
39、创建一个空集合，增加 {'x','y','z'} 三个元素。
40、删除集合 {'x','y','z'} 中的 'z' 元素，增j加元素 'w'，然后清空整个集合。
41、返回集合 {'A','D','B'} 中未出现在集合 {'D','E','C'} 中的元素（差集）。
42、返回两个集合 {'A','D','B'} 和 {'D','E','C'} 的并集。
43、返回两个集合 {'A','D','B'} 和 {'D','E','C'} 的交集。
44、返回两个集合 {'A','D','B'} 和 {'D','E','C'} 未重复的元素的集合。
45、判断两个集合 {'A','D','B'} 和 {'D','E','C'} 是否有重复元素。
46、判断集合 {'A','C'} 是否是集合 {'D','C','E','A'} 的子集。
47、去除数组 [1,2,5,2,3,4,5,'x',4,'x'] 中的重复元素。

48、返回字符串 'abCdEfg' 的全部大写、全部小写和大下写互换形式。
49、判断字符串 'abCdEfg' 是否首字母大写，字母是否全部小写，字母是否全部大写。
50、返回字符串 'this is python' 首字母大写以及字符串内每个单词首字母大写形式。
51、判断字符串 'this is python' 是否以 'this' 开头，又是否以 'python' 结尾。
52、返回字符串 'this is python' 中 'is' 的出现次数。
53、返回字符串 'this is python' 中 'is' 首次出现和最后一次出现的位置。
54、将字符串 'this is python' 切片成3个单词。
55、返回字符串 'blog.csdn.net/xufive/article/details/102946961' 按路径分隔符切片的结果。
56、将字符串 '2.72, 5, 7, 3.14' 以半角逗号切片后，再将各个元素转成浮点型或整形。
57、判断字符串 'adS12K56' 是否完全为字母数字，是否全为数字，是否全为字母？
58、将字符串 'there is python' 中的 'is' 替换为 'are'。
59、清除字符串 '\t python \n' 左侧、右侧，以及左右两侧的空白字符。
60、将三个全英文字符串（比如，'ok', 'hello', 'thank you'）分行打印，实现左对齐、右对齐和居中对齐效果。
61、将三个字符串 '15', '127', '65535' 左侧补0成同样长度。
62、将列表 ['a','b','c'] 中各个元素用'|'连接成一个字符串。
63、将字符串 'abc' 相邻的两个字母之间加上半角逗号，生成新的字符串。
64、从键盘输入手机号码，输出形如 'Mobile: 186 6677 7788' 的字符串。
65、从键盘输入年月日时分秒，输出形如 '2019-05-01 12:00:00' 的字符串。
66、给定两个浮点数 3.1415926 和 2.7182818，格式化输出字符串 'pi = 3.1416, e = 2.7183'。
67、将 0.00774592 和 356800000 格式化输出为科学计数法字符串。
68、将列表 [0,1,2,3.14,'x',None,'',list(),{5}] 中各个元素转为布尔型。
69、返回字符 'a' 和 'A' 的ASCII编码值。
70、返回ASCII编码值为 57 和 122 的字符。
71、将列表 [3,'a',5.2,4,{},9,[]] 中 大于3的整数或浮点数置为1，其余置为0。
72、将二维列表 [[1], ['a','b'], [2.3, 4.5, 6.7]] 转为 一维列表。
73、将等长的键列表和值列表转为字典。
74、数字列表求和。

答案解析：

2、求两个有序数字列表的公共元素

import random as r s=[];s1=[] for i in range(10)：s.append(r.randint(0+10*i*,*10+10*i)) s1.append(r.randint(0+10*i*,*10+10*i*))

print("s",s,"\ns1:",s1) print("两者的公共元素为：",set(s).intersection(s1)) 3、给定一个n个整型元素的列表a，其中有一个元素出现次数超过n / 2，求这个元素

class Solution：def majorityElement(self, nums) votes = 0 for num in nums：if votes == 0：x = num votes += 1 if num == x else -1 return x 这里有力扣的详细解析 https://leetcode-cn.com/problems/shu-zu-zhong-chu-xian-ci-shu-chao-guo-yi-ban-de-shu-zi-lcof/solution/mian-shi-ti-39-shu-zu-zhong-chu-xian-ci-shu-chao-3/

4、列表、元组，字典的相同点，不同点有哪些，请罗列 均为容器，列表、元组均可容纳不同的数据类型，相比于列表，元组不可原地更改（可以以切片的方式实现间接更改），可以看做一个被冻结的列表。字典以键值对作为元素，存储映射关系。 5、将元组 (1,2,3) 和集合 {4,5,6} 合并成一个列表。 print(list((1,2,3))+list({4,5,6}))

6、在列表 [1,2,3,4,5,6] 首尾分别添加整型元素 7 和 0。 a=[1,2,3,4,5,6] print(a) a.insert(0,7) a.append(0) print(a)

7、反转列表 [0,1,2,3,4,5,6,7] 。

a=[0,1,2,3,4,5,6,7] a.reverse() print(a)

8、反转列表 [0,1,2,3,4,5,6,7] 后给出中元素 5 的索引号。

a=[0,1,2,3,4,5,6,7] print(list(reversed(a)).index(5))

9、分别统计列表 [True,False,0,1,2] 中 True,False,0,1,2的元素个数，发现了什么？

python中True即1,0即False，用相应的关键字来替换

a=[True,False,0,1,2] for i in a: print(i,"在a的个数为：",a.count(i))

10、从列表 [True,1,0,‘x’,None,‘x’,False,2,True] 中删除元素‘x’。 a= [True,1,0,'x',None,'x',False,2,True] ctr=0 while 'x' in a: ctr+=1 a.remove('x') print("总共进行了%d 次remove" % ctr) print(a)

11、从列表 [True,1,0,‘x’,None,‘x’,False,2,True] 中删除索引号为4的元素。 a= [True,1,0,'x',None,'x',False,2,True] a.pop(4) print(a)

12、删除列表中索引号为奇数（或偶数）的元素。 a= [True,1,0,'x',None,'x',False,2,True] b=[] for i in range(0,len(a),2): #步长为2，从0开始 b.append(a[i]) print(b)

13、清空列表中的所有元素。 a=[True,1,0,'x',None,'x',False,2,True] del a[:] print(a)

14、对列表 [3,0,8,5,7] 分别做升序和降序排列。 a=[3,0,8,5,7] print("升序排列",sorted(a)) print("降序排列",sorted(a,reverse=True))

15、将列表 [3,0,8,5,7] 中大于 5 元素置为1，其余元素置为0。 a=[3,0,8,5,7] a=[1 if x>5 else 0 for x in a] print(a)

16、遍历列表 [‘x’,‘y’,‘z’]，打印每一个元素及其对应的索引号。 a=['x','y','z'] for i in range(len(a)): print("{}序号元素为{}" .format(i,a[i]))

17、将列表 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] 拆分为奇数组和偶数组两个列表。 a=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] l1=a[::2] l2=a[1::2] print("偶数组",l1) print("奇数组",l2)

18、分别根据每一行的首元素和尾元素大小对二维列表 [[6, 5], [3, 7], [2, 8]] 排序。 a=[[6, 5], [3, 7], [2, 8]] print("依据每行首元素升序排序",sorted(a,key=lambda x:x[0])) print("依据每行尾元素升序排序",sorted(a,key=lambda x:x[1]))

19、从列表 [1,4,7,2,5,8] 索引为3的位置开始，依次插入列表 [‘x’,‘y’,‘z’] 的所有元素。 a=[1,4,7,2,5,8] b=['x','y','z'] a=a[:3]+b+a[3:] print(a)

20、快速生成由 [5,50) 区间内的整数组成的列表。 a=list(range(5,50)) print(a)

21、若 a = [1,2,3]，令 b = a，执行 b[0] = 9， a[0]亦被改变。为何？如何避免？----讲了深COPY和浅COPY再做

对于复杂类型，如多层嵌套，import copy

b=copy.deepcopy(a)

a=[1,2,3] b=a[:] b[0]=0 print("a",a) print("b",b)

22、将列表 ['x','y','z'] 和 [1,2,3] 转成 [('x',1),('y',2),('z',3)] 的形式。 a=[1,2,3] b=['x','y','z'] print(list(zip(b,a)))

23、以列表形式返回字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中所有的键。 a={'Alice': 20, 'Beth': 18, 'Cecil': 21} print("all keys:",list(a.keys())) print("all values:",list(a.values())) print("all itemss:",list(a.items()))

24、以列表形式返回字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中所有的值。 见21题

25、以列表形式返回字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中所有键值对组成的元组。 见21题 26、向字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中追加 'David':19 键值对，更新Cecil的值为17。 a= {'Alice': 20, 'Beth': 18, 'Cecil': 21} a['David']=19 a['Cecil']=17 print(a)

27、删除字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中的Beth键后，清空该字典。 a={'Alice': 20, 'Beth': 18, 'Cecil': 21} del a['Beth'] print("删除键Beth后的字典",a) a.clear() print(a)

28、判断 David 和 Alice 是否在字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21} 中。 a={'Alice': 20, 'Beth': 18, 'Cecil': 21} keys=['David','Alice'] for key in keys: if a.get(key): print("{} 在字典中，值为{}".format(key,a[key])) else: print(key,"不在字典中")

29、遍历字典 {'Alice': 20, 'Beth': 18, 'Cecil': 21}，打印键值对。 a= {'Alice': 20, 'Beth': 18, 'Cecil': 21} for key,val in a.items(): print("key:{} value:{}".format(key,val))

30、若 a = dict()，令 b = a，执行 b.update({'x':1})， a亦被改变。为何？如何避免？----讲了深COPY和浅COPY再做 a={} b=a.copy() b.update({"x":1}) print("a",a) print("b",b)

31、以列表 ['A','B','C','D','E','F','G','H'] 中的每一个元素为键，默认值都是0，创建一个字典。 a={}.fromkeys('A B C D E F G H'.split(),0) print(a)

32、将二维结构 [['a',1],['b',2]] 和 (('x',3),('y',4)) 转成字典。 a=[['a',1],['b',2]] b=(('x',3),('y',4)) print(dict(a)) print(dict(b))

33、将元组 (1,2) 和 (3,4) 合并成一个元组。 a=(1,2) b=(3,4) print(a+b)

34、将空间坐标元组 (1,2,3) 的三个元素解包对应到变量 x,y,z。 x,y,z=(1,2,3) print(x,y,z)

35、返回元组 ('Alice','Beth','Cecil') 中 'Cecil' 元素的索引号。 a=('Alice','Beth','Cecil') print(a.index('Cecil'))

36、返回元组 (2,5,3,2,4) 中元素 2 的个数。 a=(2,5,3,2,4) print(a.count(2))

37、判断 'Cecil' 是否在元组 ('Alice','Beth','Cecil') 中。 a=('Alice','Beth','Cecil') if 'Cecil' in a:print("Cecil 在元组中") else:print("Cecil 不在元组中")

38、返回在元组 (2,5,3,7) 索引号为2的位置插入元素 9 之后的新元组。 a=(2,5,3,7) a=a[:2]+(9,)+a[2:] print(a)

39、创建一个空集合，增加 {'x','y','z'} 三个元素。 s1=set().union(['x','y','z']) print(s1)

40、删除集合 {'x','y','z'} 中的 'z' 元素，增加元素 'w'，然后清空整个集合。 s=set(['x','y','z']) s.remove('z') print("清除元素z后",s) s.clear() print(s)

41、返回集合 {'A','D','B'} 中未出现在集合 {'D','E','C'} 中的元素（差集）。 a = set(['A', 'D', 'B']) b = set(['E', 'D', 'C']) print("差集",a.difference(b)) print("交集",a.intersection(b)) print("并集",a.union(b))

42、返回两个集合 {'A','D','B'} 和 {'D','E','C'} 的并集。 见39题 43、返回两个集合 {'A','D','B'} 和 {'D','E','C'} 的交集。 见39题

44、返回两个集合 {'A','D','B'} 和 {'D','E','C'} 未重复的元素的集合。 a = set(['A', 'D', 'B']) b = set(['E', 'D', 'C']) pub=a.intersection(b)

print("两个集合未重复的元素集合为",a.union(b).difference(pub))

45、判断两个集合 {'A','D','B'} 和 {'D','E','C'} 是否有重复元素。 a = set(['A', 'D', 'B']) b = set(['E', 'D', 'C']) pub=a.intersection(b) if a.intersection(b): print("存在交集",pub) else: print("不存在交集")

46、判断集合 {'A','C'} 是否是集合 {'D','C','E','A'} 的子集。 a=set(['A','C']) b=set(['D','C','E','A'])

if a.union(b)==b: print("是子集") else: print("不是子集")

47、去除数组 [1,2,5,2,3,4,5,'x',4,'x'] 中的重复元素。 a= [1,2,5,2,3,4,5,'x',4,'x'] a=list(set(a)) print(a)

48、返回字符串 'abCdEfg' 的全部大写、全部小写和大下写互换形式。 s='abCdEfg'' print("转大写",s.upper()) print("转小写",s.lower()) print("大小写互换",s.swapcase())

49、判断字符串 'abCdEfg' 是否首字母大写，字母是否全部小写，字母是否全部大写。 s='abCdEfg' print("是否首字母大写",s.capitalize()[0]==s[0]) print("是否全是小写",s.islower()) print("是否全部大写",s.isupper())

50、返回字符串 'this is python' 首字母大写以及字符串内每个单词首字母大写形式。 s='this is python' print(s.capitalize()) print(s.title())

51、判断字符串 'this is python' 是否以 'this' 开头，又是否以 'python' 结尾。 s='this is python' print("以this开头",s.startswith("this")) print("以python结尾",s.endswith("python"))

52、返回字符串 'this is python' 中 'is' 的出现次数。 s='this is python' print("is出现次数",s.count('is'))

53、返回字符串 ‘this is python’ 中 ‘is’ 首次出现和最后一次出现的位置。 s='this is python' print("首次出现位置:{} 最后1次出现位置：{}".format(s.index('is'),s.rindex('is')))

54、将字符串 ‘this is python’ 切片成3个单词。 s='this is python' print(s.split())

55、返回字符串 ‘blog.csdn.net/xufive/article/details/102946961’ 按路径分隔符切片的结果。 s='blog.csdn.net/xufive/article/details/102946961' print(s.split('/'))

56、将字符串 ‘2.72, 5, 7, 3.14’ 以半角逗号切片后，再将各个元素转成浮点型或整形。 a='2.72, 5, 7, 3.14'.split(',') conv=lambda x:float(x) if '.' in x else int(x) print(list(map(conv,a)))

57、判断字符串 ‘adS12K56’ 是否完全为字母数字，是否全为数字，是否全为字母？ a='adS12K56' if a.isnumeric(): print('全是数字') elif a.isalpha(): print('全是字母') elif a.isalnum(): print('全是字母、数字')

58、将字符串 ‘there is python’ 中的 ‘is’ 替换为 ‘are’。 s='there is python' print(s.replace('is','are'))

59、清除字符串 ‘\t python \n’ 左侧、右侧，以及左右两侧的空白字符。 s='\t python \n' print("清楚左侧空白",s.lstrip()) print("清楚右侧空白",s.rstrip()) print("清楚两侧空白",s.strip())

60、将三个全英文字符串（比如，‘ok’, ‘hello’, ‘thank you’）分行打印，实现左对齐、右对齐和居中对齐效果。 s=['ok', 'hello', 'thank you'] l=len(s[2]) print("左对齐") for i in s:print(i.ljust(l)) print(' ’10)

print("右对齐") for i in s:print(i.rjust(l)) print(' ’10)

print("居中对齐") for i in s:print(i.center(l)) print(' ’10)

61、将三个字符串 ‘15’, ‘127’, ‘65535’ 左侧补0成同样长度。 s=['15', '127', '65535'] l=max([len(x) for x in s]) for x in s: print(x.rjust(l,'0'))

62、将列表 [‘a’,‘b’,‘c’] 中各个元素用’|'连接成一个字符串。 a=['a','b','c'] print('|'.join(a))

63、将字符串 ‘abc’ 相邻的两个字母之间加上半角逗号，生成新的字符串。 a='abc' j=', ' print(j.join(a))

64、从键盘输入手机号码，输出形如 ‘Mobile: 186 6677 7788’ 的字符串。 phone=input("输入手机号") print('Mobile: '+phone)

65、从键盘输入年月日时分秒，输出形如 ‘2019-05-01 12:00:00’ 的字符串。

2019-05-01 12:00:00

dt=input("年 月 日 时 分 秒").split() print('-'.join(dt[:3])+' '+':'.join(dt[3:]))

66、给定两个浮点数 3.1415926 和 2.7182818，格式化输出字符串 ‘pi = 3.1416, e = 2.7183’。 a=3.1415926 b=2.7182818 print("pi:{:.4f} e:{:.4f}".format(a,b))

67、将 0.00774592 和 356800000 格式化输出为科学计数法字符串。 a=0.00774592 b=356800000 print("{:e} {:e}".format(a,b))

68、将列表 [0,1,2,3.14,‘x’,None,”,list(),{5}] 中各个元素转为布尔型。 a=[0,1,2,3.14,'x',None,'',list(),{5}] a=[bool(x) for x in a] print(a)

69、返回字符 ‘a’ 和 ‘A’ 的ASCII编码值。 a=['a','A'] for i in a: print("{}的ASCII码值为{}".format(i,ord(i)))

70、返回ASCII编码值为 57 和 122 的字符。 a=[57,122] for i in a: print("{}在ASCII码表对应的字符为{}".format(i, chr(i)))

71、将列表 [3,‘a’,5.2,4,{},9,[]] 中 大于3的整数或浮点数置为1，其余置为0。 a=[3,'a',5.2,4,{},9,[]] def convert_num(x): if isinstance(x,int): return 1 if int(x)>3 else 0 elif isinstance(x,float): return 1 if float(x) > 3 else 0 else:return 0
a=list(map(convert_num,a)) print(a)

72、将二维列表 [[1], [‘a’,‘b’], [2.3, 4.5, 6.7]] 转为 一维列表。 a=[[1], ['a','b'], [2.3, 4.5, 6.7]] a=[j for x in a for j in x] print(a)

73、将等长的键列表和值列表转为字典。 a='A B C D E'.split() b=range(len(a)) dic=dict(zip(a,b)) print(dic)

74、数字列表求和。 s=list(range(10)) print(sum(s))

day4作业

1、在函数内改变函数外某个列表中第一个元素的值（和上课代码原理保持一致即可） 2、练习位置参数，keyword参数，同时练习当你传递位置参数，keyword不正确的时候，出现报错信息，理解报错原因。正确代码可以提交，错误代码也可以提交，在错误代码后面贴上报错记录，或者说这是不对的均可。（和上课代码原理保持一致即可） 3、多值参数练习，元组，字典的传参拆包练习（和上课代码原理保持一致即可） 4.设计一个类，实例化1个对象，会实现下面两种行为

需求 •一只 黄颜色 的 狗狗 叫 小黄 •具有 汪汪叫 行为 •具有 摇尾巴 行为

完成作业的同学可以刷机试题，准备复试笔试

第4题答案

class Dog: def **init**(self, name, color): self.name = name self.color = color def bark(self): print('汪汪叫')

```
def shake(self):
    print('摇尾巴')
```

day5作业

1、练习封装案例（和上课原理保持一致即可） 2、练习私有属性和私有方法（和上课原理保持一致即可） 3、练习单继承案例，包括super的使用练习（和上课原理保持一致即可）

针对作业不会写的同学，可以先根据上课的代码画画流程图，或者类图（用本子，还是电脑画图画都可以）， 然后再根据流程图，来写代码

难度作业： 4、有机试和笔试的同学，一定要积极的准备机试和笔试，不要掉以轻心，任何事情， 充分的准备是胜利的前提，周末刷刷题也挺好，而且题不会，直接看答案，搞懂了， 再自己独立去写，这就是刷题的最佳策略！

day6作业

1、练习类属性，类方法，静态方法，原理与上课一致即可 2、练习单例模式，原理与上课一致即可 3、通过try进行异常捕捉，确保输入的内容一定是一个整型数， 然后判断该整型数是否是对称数，12321就是对称数， 123321也是对称数，否则就打印不是，非整型抛异常，不是对称数抛异常

4、能够使用import来导入模块，导入包，原理与上课一致即可

做完作业的同学及时准备机试和笔试

3、通过try进行异常捕捉，确保输入的内容一定是一个整型数， 然后判断该整型数是否是对称数，12321就是对称数，123321也是对称数， 否则就不是

myException = Exception("非对称数")

结束运行，按红色正方形终止即可

while True: try: s = int(input('请输入1个数:')) if str(s) != str(s)[::-1]: raise myException print('该数是一个对称数！') except ValueError: print("非整形数") except Exception as e: print(e)

day7作业

1.完成jupyter的安装（无需提交） 2.完成文件的文本模式的读，写（与上课一致） 3.完成目录的listdir，getcwd,chdir的使用（与上课一致） 4.完成python的传参练习（与上课一致） 5、练习上课匹配单个字符，多个字符，匹配分组的正则表达式案例（与上课一致）

代码编写与上课原理一致即可，完成基础作业即可提交

难度作业： 6、完成普通文件文件的seek（代码编写与上课一致即可） 7、完成目录深度优先遍历（代码编写与上课一致即可）

day8作业

1、练习正则表达式的search，findall，sub，split，用上课的例子练习即可 2、完成二叉树前序，中序，后序，层序遍历 3、完成快速排序，堆排序

编写与上课一致，不太会写的同学，可以先根据上课的代码画画图，然后再根据自己画的图， 去逐步实现

完成作业后务必及时提交，这样可以准确把握课程进度

难度作业：

完成作业的同学，及时刷题，准备复试机试，笔试，充分准备，才能顺利上岸！