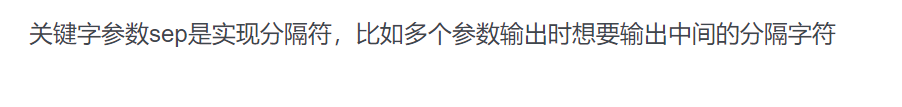
Python练习题库

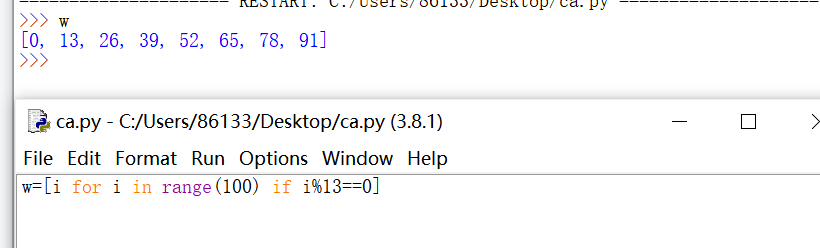
1. 填空题
2. Python标准库math中用来计算平方根的函数是\_\_\_\_\_\_\_\_\_\_。（sqrt）
3. 在Python中\_\_\_\_\_\_\_\_\_\_表示空类型。（None）
4. 列表、元组、字符串是Python的\_\_\_\_\_\_\_\_\_(有序？无序)序列。（有序）
5. 查看变量类型的Python内置函数是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（type()）
6. 查看变量内存地址的Python内置函数是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（id()）
7. 表达式[1, 2, 3]\*3的执行结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[1, 2, 3, 1, 2, 3, 1, 2, 3]）
8. list(map(str, [1, 2, 3]))的执行结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[‘1’, ‘2’, ‘3’]）
9. 已知 x = 3，并且id(x)的返回值为 496103280，那么执行语句 x += 6 之后，表达式 id(x) == 496103280 的值为\_\_\_\_\_\_\_\_\_\_\_。（False）
10. 已知 x = 3，那么执行语句 x \*= 6 之后，x的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（18）
11. 表达式“[3] in [1, 2, 3, 4]”的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（False）
12. 假设列表对象aList的值为[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]，那么切片aList[3:7]得到的值是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[6, 7, 9, 11]）
13. 使用列表推导式生成包含10个数字5的列表，语句可以写为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[5 for i in range(10)]）
14. 假设有列表a = ['name', 'age', 'sex']和b = ['Dong', 38, 'Male']，请使用一个语句将这两个列表的内容转换为字典，并且以列表a中的元素为“键”，以列表b中的元素为“值”，这个语句可以写为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（c = dict(zip(a, b))）
15. 任意长度的Python列表、元组和字符串中最后一个元素的下标为\_\_\_\_\_\_\_\_。（-1）
16. Python语句''.join(list('hello world!'))执行的结果是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'hello world!'）
17. 转义字符’\n’的含义是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（回车换行）
18. Python语句list(range(1,10,3))执行结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[1, 4, 7]）
19. 切片操作list(range(6))[::2]执行结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[0, 2, 4]）
20. 表达式 'ab' in 'acbed' 的值为\_\_\_\_\_\_\_\_。（False）
21. Python 3.x语句 print(1, 2, 3, sep=':') 的输出结果为\_\_\_\_\_\_\_\_\_\_\_\_。（1:2:3）



1. 表达式 int(4\*\*0.5) 的值为\_\_\_\_\_\_\_\_\_\_\_\_。（2）
2. 表达式 sorted([111, 2, 33], key=lambda x: -len(str(x))) 的值为\_\_\_\_\_\_\_\_\_\_\_\_。（[111, 33, 2]）

key=lambda d:d [1]是通过关键字参数来指定排序是依据元素的哪个部分来进行的，这里赋给key的是一个匿名函数（也叫lambda表达式）。sorted是升序。reverse -- 排序规则，reverse = True 降序 ， reverse = False 升序（默认）

1. 已知列表对象x = ['11', '2', '3']，则表达式 max(x) 的值为\_\_\_\_\_\_\_\_\_\_\_。（'3'）
2. 表达式 min(['11', '2', '3']) 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'11'）
3. 已知列表对象x = ['11', '2', '3']，则表达式max(x, key=len) 的值为\_\_\_\_\_\_\_\_\_\_\_。（'11'）
4. 语句 x = (3,) 执行后x的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（(3,)）
5. 语句 x = (3) 执行后x的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（3）
6. 已知 x = {1:2}，那么执行语句 x[2] = 3之后，x的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（{1: 2, 2: 3}）
7. 字典对象的\_\_\_\_\_\_\_\_\_\_\_\_\_方法返回字典中的“键-值对”列表。（items()）
8. 使用列表推导式得到100以内所有能被13整除的数的代码可以写作\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[i for i in range(100) if i%13==0]）



1. 表达式 3 \*\* 2 的值为\_\_\_\_\_\_\_\_\_。（9）
2. 表达式 3 \* 2的值为\_\_\_\_\_\_\_\_\_\_\_。（6）
3. 已知 x = [3, 5, 7]，那么执行语句 x[len(x):] = [1, 2]之后，x的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[3, 5, 7, 1, 2]）
4. 表达式 list(zip([1,2], [3,4])) 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[(1, 3), (2, 4)]）
5. 已知 x = [1, 2, 3, 2, 3]，执行语句 x.pop() 之后，x的值为\_\_\_\_\_\_\_\_\_\_\_\_\_。（[1, 2, 3, 2]）

pop() 函数用于移除列表中的一个元素（默认最后一个元素），括号里输入索引的位置。

1. 表达式 [x for x in [1,2,3,4,5] if x<3] 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[1, 2]）
2. 表达式 [index for index, value in enumerate([3,5,7,3,7]) if value == max([3,5,7,3,7])] 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[2, 4]）

------------------------------------------------------------------------------------------------------------------

1. 已知 path = r'c:\test.html'，那么表达式 path[:-4]+'htm' 的值为\_\_\_\_\_\_\_\_\_\_。（'c:\\test.htm'）
2. 表达式 '%d,%c' % (65, 65) 的值为\_\_\_\_\_\_\_\_。（'65,A'）
3. 表达式 'The first:{1}, the second is {0}'.format(65,97) 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'The first:97, the second is 65'）
4. 表达式 ':'.join('abcdefg'.split('cd')) 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'ab:efg'）
5. 表达式 isinstance('abcdefg', str) 的值为\_\_\_\_\_\_\_\_\_\_\_\_。（True）
6. 表达式 'Hello world. I like Python.'.find('python') 的值为\_\_\_\_\_\_\_\_。（-1）
7. 表达式 ','.join('a b ccc\n\n\nddd '.split()) 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'a,b,ccc,ddd'）
8. 已知 x = '123' 和 y = '456'，那么表达式 x + y 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'123456'）
9. 表达式 'abcab'.replace('a','yy') 的值为\_\_\_\_\_\_\_\_\_\_\_。（'yybcyyb'）
10. 已知 table = ''.maketrans('abcw', 'xyzc')，那么表达式 'Hellow world'.translate(table) 的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（'Helloc corld'）
11. 已知x = {'b':1, 'a':2}，那么执行语句x.update({‘a’:3, ‘d’:4})之后，表达式sorted(x.items())的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[('a', 3), ('b', 1), ('d', 4)]）
12. 已知x = list(range(20))，那么语句print(x[100:200])的输出结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[]）
13. 表达式sorted({'a':9, 'b':3, 'c':78}.values())的值为\_\_\_\_\_\_\_\_\_\_\_\_\_。（[3, 9, 78]）
14. type(1+2\*3.14)的结果是： float class
15. 阅读程序
16. 写出下面代码的执行结果。

def Join(List, sep=None):

return (sep or ',').join(List)

print(Join(['a', 'b', 'c']))

print(Join(['a', 'b', 'c'],':'))

答：

a,b,c

a:b:c

1. 若k为整数，下述while循环执行的次数为： 9

k=1000

while k>1:

print(k)

k=k//2

1. 写出下面代码的运行结果。

def Sum(a, b=3, c=5):

return sum([a, b, c])

print(Sum(a=8, c=2))

print(Sum(8))

print(Sum(8,2))

答：

13

16

15

1. 写出下列程序输出结果

i=1

while i+1:

if i>4:

print("%d"%i)

i+=1

break

print("%d"%i)

i+=1

i+=1

1. 写出下面代码的运行结果。

def Sum(\*p):

return sum(p)

print(Sum(3, 5, 8))

print(Sum(8))

print(Sum(8, 2, 10))

答：

16

8

20

1. 下面程序的执行结果是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（1）

s = 0

for i in range(1,101):

s += i

else:

print(1)

1. 下面程序的执行结果是\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（1275）

s = 0

for i in range(1,101):

s += i

if i == 50:

print(s)

break

else:

print(1)

1. 阅读下面的代码，输出结果为\_\_\_\_\_\_\_\_\_\_\_\_\_。

x = list(range(10))

for index, value in enumerate(x):

if value == 3:

x[index] = 5

else:

print(x)

答：将列表x中值为3的元素修改为5。

1. 阅读下面的代码，解释其功能：

>>> import string

>>> x = string.ascii\_letters + string.digits

>>> import random

>>> print(''.join(random.sample(x, 10)))

答：输出由英文字母大小写或数字组成的长度为10且不重复的随机字符串。

1. 下面的代码输出结果为\_\_\_\_\_\_\_\_\_\_\_\_\_。（3）#局部变量,全局变量

def demo():

x = 5

x = 3

demo()

print(x)

1. 下面程序运行的结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（[5, 6, 1, 2, 3, 4]）#序列切片

def demo(lst, k):

if k<len(lst):

return lst[k:]+lst[:k]

lst=[1,2,3,4,5,6]

demo(lst,4)

1. 下面程序运行的结果为\_\_\_\_\_ appy New Year!H\_\_\_\_\_\_\_\_\_\_。#序列切片

def foo(s):

if s=="":

return s

else:

return s[1:]+s[0]

print (foo('Happy New Year!'))

1. 下面程序运行的结果是{1:'aa',2:'bb',3:'ff',6:'dd',87: 'ee'} 。#创建字典方法

l1=[1,2,3,6,87,3]

l2=['aa','bb','cc','dd','ee','ff']

d={}

for index in range(len(l1)):

d[l1[index]]=l2[index]

print d

d=dict(zip(l1,l2)

1. 下面程序运行的结果为 [3，2] [1，2] .

list1=[1,2]

list2=list1[::]#区别:list2=list1.copy(),list2=list1

list1[0]=3

print(list1,list2)

1. 下面程序运行的结果为 恭喜，你已获得我公司的面试机会! #条件逻辑运算

age, subject,college=（24，“计算机”，”非重点”)

if (age > 25 and subject=="电子信息工程") or (college=="重点" and subject=="电子信息工程" ) or (age<=28 and subject=="计算机"):

print("恭喜，你已获得我公司的面试机会!")

else:

print("抱歉，你未达到面试要求")

1. 下面程序运行的结果为

for num in range(2,10):

if num%2 == 0:

continue

print("Find a odd numer",num)



答案：

Find a odd numer 3

Find a odd numer 5

Find a odd numer 7

Find a odd numer 9

1. 下面程序打印结果：

for i in range(1, 7):

print(' '\*2\*(6-i),end='')

for j in range(i, 0, -1):

print(j, end = " ")

print()

1. 下面程序输入>>>rev(‘I love you’)返回值为 you love I #字符串->列表

def rev1(s):

s=s.split()

s1=‘ ’.join(reversed(s))

return s1

1. 阅读程序，打印结果是 1000

def addInterest(balance,rate):

newBalance=balance\*(1+rate)

balance=newBalance

def main():

amount=1000

rate=0.05

addInterest(amount,rate)

print (amount)

main()

1. 阅读程序，打印结果是 #函数参数传值和传址(列表)

def demo(newitem,old\_list=[]):

old\_list.append(newitem)

return old\_list

def main():

print(demo(‘a’))

print(demo(‘b’))

main()

答案：['a']

['a', 'b']

1. 阅读程序，打印结果是 \*c可变长参数

**def func5(a, b, \*c):**

**print(a,b)**

**print("length of c is %d, c is " %len(c),c)**

**func5(1,2,3,4,5,6)**

答案：1 2

length of c is 4,c is (3, 4, 5, 6)

1. 阅读程序，打印结果是 (2.5, 3, 4) \*para可变长参数

def demo(\*para):

avg = sum(para)/len(para)

g = [i for i in para if i>avg]

return (avg,)+tuple(g)

print(demo(1,2,3,4))

1. 阅读程序，打印结果是 #关键字参数

def f(w=1,h=2):

print(w,h)

f()

f(w=3)

f(h=7)

f(a=3)

答案：1 2

3 2

1 7

错误

1. 阅读程序，打印结果是

def sort(number1,number2):

if number1<number2:

return number1,number2

else:

return number2,number1

n1,n2=sort(3,2)

print('n1 is ',n1)

print('n2 is ',n2)

答案：n1 is 2

n2 is 3

1. 阅读程序，打印结果是 10 60

def demo(m, n):#最大公约数、最小公倍数

if m>n:

m, n = n, m

p = m\*n

while m!=0:

r = n%m

n = m

m = r

return (n,p//n)

print( demo(20, 30))

1. 程序填空题
2. 下面代码的功能是，随机生成50个介于[1,20]之间的整数，然后统计每个整数出现频率。请把缺少的代码补全。

import random

x = [random.\_\_\_\_\_\_\_\_\_\_\_\_(1,20) for i in range(\_\_\_\_\_\_\_)]

r = dict()

for i in x:

r[i] = r.get(i, \_\_\_\_\_)+1

for k, v in r.items():

print(k, v)

答：

分别填写randint、50、0

1. 生成包含1000个随机字符的字符串，然后统计前20个高频字符，最后以15个位左对齐输出高频字符，以5个位右对齐输出个数。请把缺少的代码补全。

**import** random

x = string.ascii\_letters + string.digits + string.punctuation#大小写字母,数字,符号

y = [ **for** i **in** range(1000)]

d = dict()

**for** ch **in y**:

d[ch] = d.get(ch, 0) + 1

*i*tems=list(d.items())

items.sort( ,reverse=**True**)

**for** i **in** range(20):

word,count = items[i]

print( %(word,count))

答案：**import** string 、 random.choice(x) 、key= **lambda** x:x[1]、“%-15s%5d”

1. 用户输入5，打印如右图字符金字塔图形，请补全代码：

n=input('输入行数：')

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

print( ,end=‘ ')

for j in range(1,2\*i):

print('&', )

print()

答案：n=int(n)，' '\*(n-i)，end=''

1. 质因数分解，如输入60，则得到60 = 2 \* 2 \* 3 \* 5. 请补全代码：

x=eval(input(**"请输入小于1000的整数："**))  
k=2  
   
**while** x>1:  
 **if** :  
 print(k,end=**" "**)  
 x = x//k  
 **if** x >1:  
 print(**"\*"**,end=**" "**)   
 **else**:

答案：print(x,**"="**,end=**" "**)，x % k==0 ，k += 1

1. **验证哥德巴赫猜想：一个大于等于6的偶数可以表示为两个素数之和，请补全代码。**

**import math**

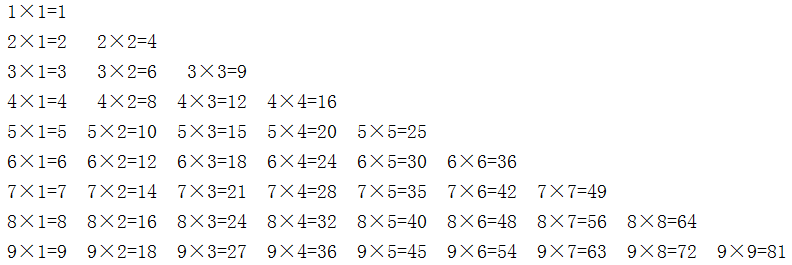
x=eval(input(**“输入一个数”**))  
**while** x<6 **or** x%2==1: #大于6的偶数  
 x=eval(input(**“输入一个数”**))  
**for** n1 **in**  :

m1=int(math.sqrt(n1)+1)  
 **for** i **in** range(2,m1): #2-sqrt(n1)  
 **if :**  
 **break  
 else**:  
   
 m2=math.ceil(math.sqrt(n2)+1)  
 **for** j **in** range(2,m2):  
 **if** n2%j==0 **and** j<n2:

**else**:  
 print(x,**'='**,n1,**'+'**,n2)

答案：range(3,x//2,2)，n1%i==0,n2=x-n1, **break**

1. 右对齐九九乘法表,请补全代码：

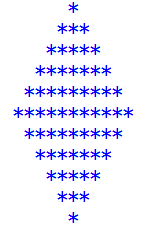


**for** i **in** range(1,10):  
   
 s1=str(i)+**'×'**+str(j)+**'='**+str(i\*j)  
 print( )

print()

答案：**for** j **in** range(1,i+1)：，**'%8s'**%(s1),end=**''**

1. 输出右边图形，请补充程序：



for i :

print((‘ \*’\*(2\*i-1)).center(30))

for i in range(6, 0, -1):

print( )

答案：in range(6)；(' \*‘(2\*i-1)).center(5\*n)

1. 下面代码判断回文串，请补全代码：

s='雾锁山头山锁雾'

low=0

high=

while low<high:

if : #倒序和顺序不一样

print(s,"不是回文")

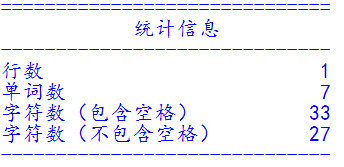
low +=1

else:

print(s,"是回文串")

答案：len(s)-1，s[low]!=s[high]，break，high -=1；

1. 输出右边图形，请补充程序：



s='''I am a teacher! You are students!'''

print('='\*30)

print('统计信息'.center(28))

print( )

item\_width=25

line=s.count('\n')+1

word=s.split()

char=list(s) #包含空格、标点符号

w\_num=0

for w in word:

w\_num+=

print('%-\*s %5d'%(item\_width-3,'行数',line))

print('%-\*s %5d'%( ,'单词数',len(word)))

print('%-\*s %5d'%(item\_width-10,'字符数（包含空格）', ))

print('%-\*s %5d'%(item\_width-11,'字符数（不包含空格）',w\_num))

print('-'\*30)

答案：'-'\*30；len(w)；item\_width-4；len(char)

1. 输出右边图形，请补充程序：



def demo(t):

print([1])

line = [1,1]

for i in range(2,t):

for j in range(0,len(line)-1):

r.append( )

line = [1]+r+[1]

print(line)

demo( )

答案：print([1,1])，r = []，line[j]+line[j+1]，10

1. 编程题
2. 编写程序，在D盘根目录下创建一个文本文件test.txt，并向其中写入字符串hello world。

答：

fp = open(r’D:\test.txt’, ‘a+’)

print(‘hello world’, file=fp)

fp.close()

1. 编写程序，生成一个包含20个随机整数的列表，然后对其中偶数下标的元素进行降序排列，奇数下标的元素不变。（提示：使用切片。）

答：

import random

x = [random.randint(0,100) for i in range(20)]

print(x)

y = x[::2]

y.sort(reverse=True)

x[::2] = y

print(x)

1. 编写函数，判断一个数字是否为素数，是则返回字符串YES，否则返回字符串NO。再编写测试函数。

答：

import math

def IsPrime(v):

n = int(math.sqrt(v)+1)

for i in range(2,n):

if v%i==0:

return 'No'

else:

return 'Yes'

print(IsPrime(77))

1. 编写函数，模拟Python内置函数sorted(lst)。

答：

def sorted(v):

t = v[::]

r = []

while t:

tt = min(t)

r.append(tt)

t.remove(tt)

return r

print(sorted([1,3,2,4])

1. 编写程序，生成包含20个随机数的列表，然后将前10个元素升序排列，后10个元素降序排列，并输出结果。

答：

import random

x = [random.randint(0,100) for i in range(20)]

print(x)

y = x[0:10]

y.sort()

x[0:10] = y

y = x[10:20]

y.sort(reverse=True)

x[10:20] = y

|  |  |
| --- | --- |
| x | y |
| x<0 | 0 |
| 0<=x<5 | x |
| 5<=x<10 | 3x-5 |
| 10<=x<20 | 0.5x-2 |
| 20<=x | 0 |

print(x)

1. 编写程序，运行后用户输入4位整数作为年份，判断其是否为闰年。如果年份能被400整除，则为闰年；如果年份能被4整除但不能被100整除也为闰年。

答：

x = input('Please input an integer of 4 digits meaning the year:')

x = eval(x)

if x%400==0 or (x%4==0 and not x%100==0):

print('Yes')

else:

print('No')

1. 编写程序，实现分段函数计算，如下表所示。

答：

x = input('Please input x:')

x = eval(x)

if x<0 or x>=20:

print(0)

elif 0<=x<5:

print(x)

elif 5<=x<10:

print(3\*x-5)

elif 10<=x<20:

print(0.5\*x-2)

1. 已知有一个包含一些同学成绩的字典，计算成绩的最高分、最低分、平均分，并查找最高分同学，最后打印最高分、最低分、平均分、最高分同学。

scores = {"Zhang San": 45, "Li Si": 78, "Wang Wu": 40, "Zhou Liu": 96, "Zhao Qi": 65, "Sun Ba": 90, "Zheng Jiu": 78, "Wu Shi": 99, "Dong Shiyi": 60}

highest = max(scores.values())

lowest = min(scores.values())

average = sum(scores.values())/len(scores)

highestPerson = [name for name, score in scores.items() if / score == highest] #分行

print(highest,lowest,average,highestPerson)

1. 统计不及格的学生及成绩，要求：给定一些学生姓名和成绩；询问用户是否还需要输入更多的学生姓名和成绩，用户回答是就继续输入，继续询问；打印所有补考学生名单和成绩。

d=[["张三",76],["李四",45],["王五",83],["郑六",66],['a',54],['c',68],['b',89]]

s1=input("输入数据Y or N:").strip()#去掉前后空格

while s1 == 'Y':

s2=input("请输入学生及成绩（空格隔开）")

name,grade=s2.split()

d.append([name,grade])

#d[name]=int(grade)

s1=input("继续输入Y or N:").strip()

else:

print(d)

print("不及格的名单:")

for name,grade in d:

if float(grade) < 60:

print("姓名：",name,"\t","成绩：",grade)

1. 由用户一个一个输入一个数值型列表，然后将每个元素逐一打印。

lst=[]

n=eval(input(”请输入数字个数：”))

for i in range(n):

number=eval(input(”请输入数字：”))

lst.append(number)

for value in lst:

print(value,end=’ ’)

1. **序列求和，n通过用户输入:**



n=int(input(**“请输入N的值:”**))  
s=0  
fh=1  
**for** i **in** range(1,n+1):  
 s +=fh\*1/i # 求和语句   
 fh=-fh #正负交替

print(**"1-1/2+1/3-1/4...+(-1)^(n-1)/n="**,s,i)

1. 生成一个包含50个两位随机整数的列表，然后删除其中的所有奇数(提示从后向前删)

lst=[random.randint(1,100) for i in range(50)]

print('50个随机两位整数：',lst)

k=len(lst)-1

for i in range(k,-1,-1):#从后往前删，减少列表元素移动

if lst[i] % 2==1:

lst.pop(i)

print("偶数为",lst)

1. 恺撒（kaisa）密码：原文：ABCDEFGHIJKLMNOPQRSTUVWXYZ，对应的密文为：

DEFGHIJKLMNOPQRSTUVWXYZABC，请设计程序实现之。

plaincode=input('请输入明文：')

for p in plaincode:

if 'a'<=p<='z':

c=chr(ord('a')+(ord(p)-ord('a')+3)%26)

#ord是字符编码的值，都减掉ord('A')才可以得到字符的顺序0-25

print(c,end='')

elif 'A'<=p<='Z' :

c=chr(ord('A')+(ord(p)-ord('A')+3)%26)

print(c,end='')

else:

print(p,end='')

1. 如果一个字符串从前往后和从后往前读时是一样的，那么这个字符串就是回文串。请编写一个函数判断是否为回文串，同时编写测试代码检验回文串。

def isPalindrome(s)

if s==s[::-1]: #切片操作，倒序和顺序一样

return True

else:

return False

s=input('input a string:')

if isPalindrome(s):

print(‘%s is a palindrome!’%s)

else:

print(‘%s is not a palindrome!’%s)

1. 输入一串字符，统计每个字符数，用字典输出.

s=input("输入一行字符串或句子：")

char\_counts={}

for char in s:

char\_counts[char]= char\_counts.get(char,0)+1

print(char\_counts)

1. 生成包含100个随机字符的字符串，然后统计前20个高频字符.

import string,random

str1=string.ascii\_letters +string.digits

str1=str1.lower()

str2=[random.choice (str1) for i in range(100)]

print(str2)

str\_count={}

for char in str2:

str\_count[char]=str\_count.get(char,0)+1

print(str\_count)

items=list(str\_count.items())

items.sort(key=lambda x:x[1],reverse=True) #以数量降序

for i in range(20):

word,count = items[i]

print("%-15s %5d"%(word,count)) #-表示左对齐，s表字符，d表示整数

1. 统计”e:\HarryPotterSorcerStone.txt”文件中的英文单词及其数量,打印出单词及其个数

filename =r'HarryPotterSorcerStone.txt'

infile = open(filename, "r") # Open the file

txt=infile.read()

words=txt.split() #以空格分割单词

words\_counts={}

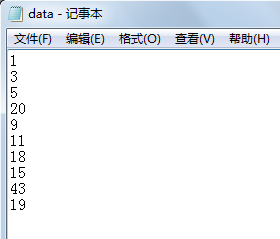
for word in words:

words\_counts[word]=words\_counts.get(word,0)+1

print(words\_counts)

1. 读取文本文件data.txt中所有整数，将其排序后写入文本文件data\_asc.txt中.

with open('data.txt','r') as fp:



data=fp.readlines() #文件全部读完返回字符列表

print(‘字符列表’,data)

data=[int(line.strip()) for line in data] #列表的数据变成整数

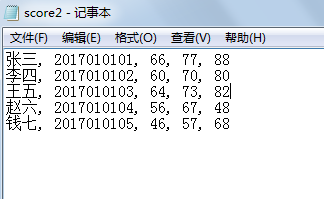
data.sort()

data=[str(i)+'\n‘ for i in data] #数字后加换行符

with open(‘data\_asc.txt’,'w') as fp:

fp.writelines(data) #一次全部写入

1. 编程，如图，在文件score.txt中写入五名学生的姓名、学号和3门考试课的成绩，然后将所有两门以上（含两门）课程不及格的学生信息输出到文件bad.txt、其他学生信息输出到pass.txt。



fw=open(r'score2.txt','w')

s=['张三,',' 2017010101,',' 66,',' 77,',' 88','\n']

fw.writelines(s)

fw.write('李四, 2017010102, 60, 70, 80\n')

fw.write('王五, 2017010103, 64, 73, 82\n')

fw.write('赵六, 2017010104, 56, 67, 48\n')

fw.write('钱七, 2017010105, 46, 57, 68\n')

fw.close()

f1=open(r'score2.txt','r')

f2=open(r'bad2.txt','w')

f3=open(r'pass2.txt','w')

lines=f1.readlines()

for line in lines:

data=list(line.split(','))

print(data)#['张三', ' 2017010101', ' 66', ' 77', ' 88\n']

k=0

for i in data[2:]:

if int(i)<60:

k+=1

if k>=2: #两门以上（含两门）课程不及格

f2.write(line)

else:

f3.write(line)

f1.close();f2.close();f3.close()