# Python3 100例

> 原题地址：

> http://www.runoob.com/python/python-100-examples.html

#### 实例001：数字组合

\*\*题目：\*\*有四个数字：1、2、3、4，能组成多少个互不相同且无重复数字的三位数？各是多少？

\*\*程序分析：\*\*遍历全部可能，把有重复的剃掉。

```

total=0

for i in range(1,5):

for j in range(1,5):

for k in range(1,5):

if ((i!=j)and(j!=k)and(k!=i)):

print(i,j,k)

total+=1

print(total)

```

\*\*简便方法：\*\*用itertools中的permutations即可。

```

import itertools

sum2=0

a=[1,2,3,4]

for i in itertools.permutations(a,3):

print(i)

sum2+=1

print(sum2)

```

#### 实例002：“个税计算”

\*\*题目：\*\*企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万到60万之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%，高于100万元时，超过100万元的部分按1%提成，从键盘输入当月利润I，求应发放奖金总数？

\*\*程序分析：\*\*分区间计算即可。

```

profit=int(input('Show me the money: '))

bonus=0

thresholds=[100000,100000,200000,200000,400000]

rates=[0.1,0.075,0.05,0.03,0.015,0.01]

for i in range(len(thresholds)):

if profit<=thresholds[i]:

bonus+=profit\*rates[i]

profit=0

break

else:

bonus+=thresholds[i]\*rates[i]

profit-=thresholds[i]

bonus+=profit\*rates[-1]

print(bonus)

```

#### 实例003：完全平方数

\*\*题目：\*\*一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，请问该数是多少？

\*\*程序分析：\*\*因为168对于指数爆炸来说实在太小了，所以可以直接省略数学分析，用最朴素的方法来获取上限:

```

n=0

while (n+1)\*\*2-n\*n<=168:

n+=1

print(n+1)

----------

85

```

思路是：最坏的结果是n的平方与(n+1)的平方刚好差168，由于是平方的关系，不可能存在比这更大的间隙。

至于判断是否是完全平方数，最简单的方法是：平方根的值小数为0即可。

结合起来：

```

n=0

while (n+1)\*\*2-n\*n<=168:

n+=1

for i in range((n+1)\*\*2):

if i\*\*0.5==int(i\*\*0.5) and (i+168)\*\*0.5==int((i+168)\*\*0.5):

print(i-100)

```

#### 实例004：这天第几天

\*\*题目：\*\*输入某年某月某日，判断这一天是这一年的第几天？

\*\*程序分析：\*\*特殊情况，闰年时需考虑二月多加一天：

```

def isLeapYear(y):

return (y%400==0 or (y%4==0 and y%100!=0))

DofM=[0,31,28,31,30,31,30,31,31,30,31,30]

res=0

year=int(input('Year:'))

month=int(input('Month:'))

day=int(input('day:'))

if isLeapYear(year):

DofM[2]+=1

for i in range(month):

res+=DofM[i]

print(res+day)

```

#### 实例005：三数排序

\*\*题目：\*\*输入三个整数x,y,z，请把这三个数由小到大输出。

\*\*程序分析：\*\*练练手就随便找个排序算法实现一下，偷懒就直接调函数。

```

raw=[]

for i in range(3):

x=int(input('int%d: '%(i)))

raw.append(x)

for i in range(len(raw)):

for j in range(i,len(raw)):

if raw[i]>raw[j]:

raw[i],raw[j]=raw[j],raw[i]

print(raw)

raw2=[]

for i in range(3):

x=int(input('int%d: '%(i)))

raw2.append(x)

print(sorted(raw2))

```

#### 实例006：斐波那契数列

\*\*题目：\*\*斐波那契数列。

\*\*程序分析：\*\*斐波那契数列（Fibonacci sequence），从1,1开始，后面每一项等于前面两项之和。图方便就递归实现，图性能就用循环。

```

# 递归实现

def Fib(n):

return 1 if n<=2 else Fib(n-1)+Fib(n-2)

print(Fib(int(input())))

# 朴素实现

target=int(input())

res=0

a,b=1,1

for i in range(target-1):

a,b=b,a+b

print(a)

```

#### 实例007：copy

\*\*题目：\*\*将一个列表的数据复制到另一个列表中。

\*\*程序分析：\*\*使用列表[:]，拿不准可以调用copy模块。

```

import copy

a = [1,2,3,4,['a','b']]

b = a # 赋值

c = a[:] # 浅拷贝

d = copy.copy(a) # 浅拷贝

e = copy.deepcopy(a) # 深拷贝

a.append(5)

a[4].append('c')

print('a=',a)

print('b=',b)

print('c=',c)

print('d=',d)

print('e=',e)

============ RESTART: F:\PyWorkspace\Python100\100examples\007.py ============

a= [1, 2, 3, 4, ['a', 'b', 'c'], 5]

b= [1, 2, 3, 4, ['a', 'b', 'c'], 5]

c= [1, 2, 3, 4, ['a', 'b', 'c']]

d= [1, 2, 3, 4, ['a', 'b', 'c']]

e= [1, 2, 3, 4, ['a', 'b']]

```

#### 实例008:九九乘法表

\*\*题目：\*\*输出 9\*9 乘法口诀表。

\*\*程序分析：\*\*分行与列考虑，共9行9列，i控制行，j控制列。

```

for i in range(1,10):

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

print('%d\*%d=%2ld '%(i,j,i\*j),end='')

print()

```

#### 实例009：暂停一秒输出

\*\*题目：\*\*暂停一秒输出。

\*\*程序分析：\*\*使用 time 模块的 sleep() 函数。

```

import time

for i in range(4):

print(str(int(time.time()))[-2:])

time.sleep(1)

```

#### 实例010：给人看的时间

\*\*题目：\*\*暂停一秒输出，并格式化当前时间。

\*\*程序分析：\*\*同009.

```

import time

for i in range(4):

print(time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time())))

time.sleep(1)

```

#### 实例011：养兔子

\*\*题目：\*\*有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

\*\*程序分析：\*\*我认为原文的解法有点扯，没有考虑3个月成熟的问题，人家还是婴儿怎么生孩子？考虑到三个月成熟，可以构建四个数据，其中：一月兔每个月长大成为二月兔，二月兔变三月兔，三月兔变成年兔，成年兔（包括新成熟的三月兔）生等量的一月兔。

```

month=int(input('繁殖几个月？： '))

month\_1=1

month\_2=0

month\_3=0

month\_elder=0

for i in range(month):

month\_1,month\_2,month\_3,month\_elder=month\_elder+month\_3,month\_1,month\_2,month\_elder+month\_3

print('第%d个月共'%(i+1),month\_1+month\_2+month\_3+month\_elder,'对兔子')

print('其中1月兔：',month\_1)

print('其中2月兔：',month\_2)

print('其中3月兔：',month\_3)

print('其中成年兔：',month\_elder)

```

#### 实例012:100到200的素数

\*\*题目：\*\*判断101-200之间有多少个素数，并输出所有素数。

\*\*程序分析：\*\*判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除，则表明此数不是素数，反之是素数。

```

import math

for i in range(100,200):

flag=0

for j in range(2,round(math.sqrt(i))+1):

if i%j==0:

flag=1

break

if flag:

continue

print(i)

print('\nSimplify the code with "else"\n')

for i in range(100,200):

for j in range(2,round(math.sqrt(i))+1):

if i%j==0:

break

else:

print(i)

```

#### 实例013：所有水仙花数

\*\*题目：\*\*打印出所有的"水仙花数"，所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。例如：153是一个"水仙花数"，因为153=1的三次方＋5的三次方＋3的三次方。

\*\*程序分析：\*\*利用for循环控制100-999个数，每个数分解出个位，十位，百位。

```

for i in range(100,1000):

s=str(i)

one=int(s[-1])

ten=int(s[-2])

hun=int(s[-3])

if i == one\*\*3+ten\*\*3+hun\*\*3:

print(i)

```

#### 实例014：分解质因数

\*\*题目：\*\*将一个整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5。

\*\*程序分析：\*\*根本不需要判断是否是质数，从2开始向数本身遍历，能整除的肯定是最小的质数。

```

target=int(input('输入一个整数：'))

print(target,'= ',end='')

if target<0:

target=abs(target)

print('-1\*',end='')

flag=0

if target<=1:

print(target)

flag=1

while True:

if flag:

break

for i in range(2,int(target+1)):

if target%i==0:

print("%d"%i,end='')

if target==i:

flag=1

break

print('\*',end='')

target/=i

break

```

#### 实例015：分数归档

\*\*题目：\*\*利用条件运算符的嵌套来完成此题：学习成绩>=90分的同学用A表示，60-89分之间的用B表示，60分以下的用C表示。

\*\*程序分析：\*\*用条件判断即可。

```

points=int(input('输入分数：'))

if points>=90:

grade='A'

elif points<60:

grade='C'

else:

grade='B'

print(grade)

```

#### 实例016：输出日期

\*\*题目：\*\*输出指定格式的日期。

\*\*程序分析：\*\*使用 datetime 模块。

```

import datetime

print(datetime.date.today())

print(datetime.date(2333,2,3))

print(datetime.date.today().strftime('%d/%m/%Y'))

day=datetime.date(1111,2,3)

day=day.replace(year=day.year+22)

print(day)

```

#### 实例017：字符串构成

\*\*题目：\*\*输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

\*\*程序分析：\*\*利用 while 或 for 语句,条件为输入的字符不为 '\n'。

```

string=input("输入字符串：")

alp=0

num=0

spa=0

oth=0

for i in range(len(string)):

if string[i].isspace():

spa+=1

elif string[i].isdigit():

num+=1

elif string[i].isalpha():

alp+=1

else:

oth+=1

print('space: ',spa)

print('digit: ',num)

print('alpha: ',alp)

print('other: ',oth)

```

#### 实例018：复读机相加

\*\*题目：\*\*求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)，几个数相加由键盘控制。

\*\*程序分析：\*\*用字符串解决。

```

a=input('被加数字：')

n=int(input('加几次？：'))

res=0

for i in range(n):

res+=int(a)

a+=a[0]

print('结果是：',res)

```

#### 实例019：完数

\*\*题目：\*\*一个数如果恰好等于它的因子之和，这个数就称为"完数"。例如6=1＋2＋3.编程找出1000以内的所有完数。

\*\*程序分析：\*\*将每一对因子加进集合，在这个过程中已经自动去重。最后的结果要求不计算其本身。

```

def factor(num):

target=int(num)

res=set()

for i in range(1,num):

if num%i==0:

res.add(i)

res.add(num/i)

return res

for i in range(2,1001):

if i==sum(factor(i))-i:

print(i)

```

#### 实例020：高空抛物

\*\*题目：\*\*一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第10次落地时，共经过多少米？第10次反弹多高？

\*\*程序分析：\*\*无

```

high=200.

total=100

for i in range(10):

high/=2

total+=high

print(high/2)

print('总长：',total)

```

#### 实例021：猴子偷桃

\*\*题目：\*\*猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

\*\*程序分析：\*\*按规则反向推断：猴子有一个桃子，他偷来一个桃子，觉得不够又偷来了与手上等量的桃子，一共偷了9天。

```

peach=1

for i in range(9):

peach=(peach+1)\*2

print(peach)

```

#### 实例022：比赛对手

\*\*题目：\*\*两个乒乓球队进行比赛，各出三人。甲队为a,b,c三人，乙队为x,y,z三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a说他不和x比，c说他不和x,z比，请编程序找出三队赛手的名单。

\*\*程序分析：\*\*找到条件下不重复的三个对手即可。

```

a=set(['x','y','z'])

b=set(['x','y','z'])

c=set(['x','y','z'])

c-=set(('x','z'))

a-=set('x')

for i in a:

for j in b:

for k in c:

if len(set((i,j,k)))==3:

print('a:%s,b:%s,c:%s'%(i,j,k))

```

#### 实例023：画菱形

\*\*题目：\*\*打印出如下图案（菱形）:

> &nbsp;&nbsp;&nbsp;&nbsp;\\*

> &nbsp;&nbsp;&nbsp;\\*\\*\\*

> &nbsp;&nbsp;\\*\\*\\*\\*\\*

> &nbsp;\\*\\*\\*\\*\\*\\*\\*

> &nbsp;&nbsp;\\*\\*\\*\\*\\*

> &nbsp;&nbsp;&nbsp;\\*\\*\\*

> &nbsp;&nbsp;&nbsp;&nbsp;\\*

\*\*程序分析：\*\*递归调用即可。

```

def draw(num):

a="\*"\*(2\*(4-num)+1)

print(a.center(9,' '))

if num!=1:

draw(num-1)

print(a.center(9,' '))

draw(4)

```

#### 实例024：斐波那契数列II

\*\*题目：\*\*有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...求出这个数列的前20项之和。

\*\*程序分析：\*\*就是斐波那契数列的后一项除以前一项。

```

a = 2.0

b = 1.0

s = 0

for n in range(1,21):

s += a / b

a,b = a + b,a

print (s)

```

#### 实例025： 阶乘求和

\*\*题目：\*\*求1+2!+3!+...+20!的和。

\*\*程序分析：\*\*1+2!+3!+...+20!=1+2(1+3(1+4(...20(1))))

```

res=1

for i in range(20,1,-1):

res=i\*res+1

print(res)

```

#### 实例026：递归求阶乘

\*\*题目：\*\*利用递归方法求5!。

\*\*程序分析：\*\*递归调用即可。

```

def factorial(n):

return n\*factorial(n-1) if n>1 else 1

print(factorial(5))

```

#### 实例027：递归输出

\*\*题目：\*\*利用递归函数调用方式，将所输入的5个字符，以相反顺序打印出来。

\*\*程序分析：\*\*递归真是蠢方法。

```

def rec(string):

if len(string)!=1:

rec(string[1:])

print(string[0],end='')

rec(input('string here:'))

```

#### 实例028：递归求等差数列

\*\*题目：\*\*有5个人坐在一起，问第五个人多少岁？他说比第4个人大2岁。问第4个人岁数，他说比第3个人大2岁。问第三个人，又说比第2人大两岁。问第2个人，说比第一个人大两岁。最后问第一个人，他说是10岁。请问第五个人多大？

\*\*程序分析：\*\*就一等差数列。

```

def age(n):

if n==1:

return 10

return 2+age(n-1)

print(age(5))

```

#### 实例029：反向输出

\*\*题目：\*\*给一个不多于5位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。

\*\*程序分析：\*\*学会分解出每一位数,用字符串的方法总是比较省事。

```

n=int(input('输入一个正整数：'))

n=str(n)

print('%d位数'%len(n))

print(n[::-1])

```

#### 实例030：回文数

\*\*题目：\*\*一个5位数，判断它是不是回文数。即12321是回文数，个位与万位相同，十位与千位相同。

\*\*程序分析：\*\*用字符串比较方便,就算输入的不是数字都ok。

```

n=input("随便你输入啥啦：")

a=0

b=len(n)-1

flag=True

while a<b:

if n[a]!=n[b]:

print('不是回文串')

flag=False

break

a,b=a+1,b-1

if flag:

print('是回文串')

```

#### 实例031：字母识词

\*\*题目：\*\*请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续判断第二个字母。

\*\*程序分析：\*\*这里用字典的形式直接将对照关系存好。

```

weekT={'h':'thursday',

'u':'tuesday'}

weekS={'a':'saturday',

'u':'sunday'}

week={'t':weekT,

's':weekS,

'm':'monday',

'w':'wensday',

'f':'friday'}

a=week[str(input('请输入第一位字母:')).lower()]

if a==weekT or a==weekS:

print(a[str(input('请输入第二位字母:')).lower()])

else:

print(a)

```

#### 实例032：反向输出II

\*\*题目：\*\*按相反的顺序输出列表的值。

\*\*程序分析：\*\*无。

```

a = ['one', 'two', 'three']

print(a[::-1])

```

#### 实例033：列表转字符串

\*\*题目：\*\*按逗号分隔列表。

\*\*程序分析：\*\*无。

```

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

print(','.join(str(n) for n in L))

```

#### 实例034：调用函数

\*\*题目：\*\*练习函数调用。

\*\*程序分析：\*\*无。

```

def hello():

print('Hello World!')

def helloAgain():

for i in range(2):

hello()

if \_\_name\_\_=='\_\_main\_\_':

helloAgain()

```

#### 实例035：设置输出颜色

\*\*题目：\*\*文本颜色设置。

\*\*程序分析：\*\*无。

```

class bcolors:

HEADER = '\033[95m'

OKBLUE = '\033[94m'

OKGREEN = '\033[92m'

WARNING = '\033[93m'

FAIL = '\033[91m'

ENDC = '\033[0m'

BOLD = '\033[1m'

UNDERLINE = '\033[4m'

print(bcolors.WARNING + "警告的颜色字体?" + bcolors.ENDC)

```

#### 实例036：算素数

\*\*题目：\*\*求100之内的素数。

\*\*程序分析：\*\*用else执行for循环的奖励代码（如果for是正常完结，非break）。

```

lo=int(input('下限：'))

hi=int(input('上限：'))

for i in range(lo,hi+1):

if i > 1:

for j in range(2,i):

if (i % j) == 0:

break

else:

print(i)

```

#### 实例037：排序

\*\*题目：\*\*对10个数进行排序。

\*\*程序分析：\*\*同实例005。

```

raw=[]

for i in range(10):

x=int(input('int%d: '%(i)))

raw.append(x)

for i in range(len(raw)):

for j in range(i,len(raw)):

if raw[i]>raw[j]:

raw[i],raw[j]=raw[j],raw[i]

print(raw)

```

#### 实例038：矩阵对角线之和

\*\*题目：\*\*求一个3\*3矩阵主对角线元素之和。

\*\*程序分析：\*\*无。

```

mat=[[1,2,3],

[3,4,5],

[4,5,6]

]

res=0

for i in range(len(mat)):

res+=mat[i][i]

print(res)

```

#### 实例039：有序列表插入元素

\*\*题目：\*\*有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中。

\*\*程序分析：\*\*首先判断此数是否大于最后一个数，然后再考虑插入中间的数的情况，插入后此元素之后的数，依次后移一个位置。

```

lis=[1,10,100,1000,10000,100000]

n=int(input('insert a number: '))

lis.append(n)

for i in range(len(lis)-1):

if lis[i]>=n:

for j in range(i,len(lis)):

lis[j],lis[-1]=lis[-1],lis[j]

break

print(lis)

```

#### 实例040：逆序列表

\*\*题目：\*\*将一个数组逆序输出。

\*\*程序分析：\*\*依次交换位置，或者直接调用reverse方法。

```

lis=[1,10,100,1000,10000,100000]

for i in range(int(len(lis)/2)):

lis[i],lis[len(lis)-1-i]=lis[len(lis)-1-i],lis[i]

print('第一种实现：')

print(lis)

lis=[1,10,100,1000,10000,100000]

print('第二种实现：')

lis.reverse()

print(lis)

```

#### 实例041：类的方法与变量

\*\*题目：\*\*模仿静态变量的用法。

\*\*程序分析：\*\*构造类，了解类的方法与变量。

```

def dummy():

i=0

print(i)

i+=1

class cls:

i=0

def dummy(self):

print(self.i)

self.i+=1

a=cls()

for i in range(50):

dummy()

a.dummy()

```

#### 实例042：变量作用域

\*\*题目：\*\*学习使用auto定义变量的用法。

\*\*程序分析：\*\*python中的变量作用域。

```

i=0

n=0

def dummy():

i=0

print(i)

i+=1

def dummy2():

global n

print(n)

n+=1

print('函数内部的同名变量')

for j in range(20):

print(i)

dummy()

i+=1

print('global声明同名变量')

for k in range(20):

print(n)

dummy2()

n+=10

```

#### 实例043：作用域、类的方法与变量

\*\*题目：\*\*模仿静态变量(static)另一案例。

\*\*程序分析：\*\*综合实例041和实例042。

```

class dummy:

num=1

def Num(self):

print('class dummy num:',self.num)

print('global num: ',num)

self.num+=1

n=dummy()

num=1

for i in range(5):

num\*=10

n.Num()

```

#### 实例044：矩阵相加

\*\*题目：\*\*计算两个矩阵相加。

\*\*程序分析：\*\*创建一个新的矩阵，使用 for 迭代并取出 X 和 Y 矩阵中对应位置的值，相加后放到新矩阵的对应位置中。

```

X = [[12,7,3],

[4 ,5,6],

[7 ,8,9]]

Y = [[5,8,1],

[6,7,3],

[4,5,9]]

res=[[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(res)):

for j in range(len(res[0])):

res[i][j]=X[i][j]+Y[i][j]

print(res)

```

#### 实例045：求和

\*\*题目：\*\*统计 1 到 100 之和。

\*\*程序分析：\*\*无

```

res=0

for i in range(1,101):

res+=i

print(res)

```

#### 实例046：打破循环

\*\*题目：\*\*求输入数字的平方，如果平方运算后小于 50 则退出。

\*\*程序分析：\*\*无

```

while True:

try:

n=float(input('输入一个数字：'))

except:

print('输入错误')

continue

dn=n\*\*2

print('其平方为：',dn)

if dn<50:

print('平方小于50，退出')

break

```

#### 实例047：函数交换变量

\*\*题目：\*\*两个变量值用函数互换。

\*\*程序分析：\*\*无

```

def exc(a,b):

return (b,a)

a=0

b=10

a,b=exc(a,b)

print(a,b)

```

#### 实例048：数字比大小

\*\*题目：\*\*数字比较。

\*\*程序分析：\*\*无

```

a=int(input('a='))

b=int(input('b='))

if a<b:

print('a<b')

elif a>b:

print('a>b')

else:

print('a=b')

```

#### 实例049：lambda

\*\*题目：\*\*使用lambda来创建匿名函数。

\*\*程序分析：\*\*无

```

Max=lambda x,y:x\*(x>=y)+y\*(y>x)

Min=lambda x,y:x\*(x<=y)+y\*(y<x)

a=int(input('1:'))

b=int(input('2:'))

print(Max(a,b))

print(Min(a,b))

```

#### 实例050：随机数

\*\*题目：\*\*输出一个随机数。

\*\*程序分析：\*\*使用 random 模块。

```

import random

print(random.uniform(10,20))

```

#### 实例051：按位与

\*\*题目：\*\*学习使用按位与 & 。

\*\*程序分析：\*\*0&0=0; 0&1=0; 1&0=0; 1&1=1。

```

a=0o77

print(a)

b=a&3

print(b)

b=b&7

print(b)

```

#### 实例052：按位或

\*\*题目：\*\*学习使用按位或 | 。

\*\*程序分析：\*\*0|0=0; 0|1=1; 1|0=1; 1|1=1

```

a=0o77

print(a|3)

print(a|3|7)

```

#### 实例053：按位异或

\*\*题目：\*\*学习使用按位异或 ^ 。

\*\*程序分析：\*\*0^0=0; 0^1=1; 1^0=1; 1^1=0

```

a=0o77

print(a^3)

print(a^3^7)

```

#### 实例054：位取反、位移动

\*\*题目：\*\*取一个整数a从右端开始的4〜7位。

\*\*程序分析：\*\*可以这样考虑：

(1)先使a右移4位。

(2)设置一个低4位全为1,其余全为0的数。可用~(~0<<4)

(3)将上面二者进行&运算。

```

a=int(input('输入一个数字: '))

b=0 # 0

b=~b # 1

b=b<<4 # 10000

b=~b # 1111

c=a>>4

d=c&b

print('a:',bin(a))

print('b:',bin(b))

print('c:',bin(c))

print('d:',bin(d))

```

#### 实例055：按位取反

\*\*题目：\*\*学习使用按位取反~。

\*\*程序分析：\*\*~0=1; ~1=0;

```

print(~234)

print(~~234)

```

#### 实例056：画圈

\*\*题目：\*\*画图，学用circle画圆形。

\*\*程序分析：\*\*无。

```

from tkinter import \*

canvas=Canvas(width=800,height=600,bg='yellow')

canvas.pack(expand=YES,fill=BOTH)

k=1

j=1

for i in range(26):

canvas.create\_oval(310-k,250-k,310+k,250+k,width=1)

k+=j

j+=0.3

mainloop()

```

#### 实例057：画线

\*\*题目：\*\*画图，学用line画直线。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

from tkinter import \*

canvas = Canvas(width=300, height=300, bg='green')

canvas.pack(expand=YES, fill=BOTH)

x0 = 263

y0 = 263

y1 = 275

x1 = 275

for i in range(19):

canvas.create\_line(x0,y0,x0,y1, width=1, fill='red')

x0 = x0 - 5

y0 = y0 - 5

x1 = x1 + 5

y1 = y1 + 5

x0 = 263

y1 = 275

y0 = 263

for i in range(21):

canvas.create\_line(x0,y0,x0,y1,fill = 'red')

x0 += 5

y0 += 5

y1 += 5

mainloop()

```

#### 实例058：画矩形

\*\*题目：\*\*画图，学用rectangle画方形。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

from tkinter import \*

root = Tk()

root.title('Canvas')

canvas = Canvas(root,width = 400,height = 400,bg = 'yellow')

x0 = 263

y0 = 263

y1 = 275

x1 = 275

for i in range(19):

canvas.create\_rectangle(x0,y0,x1,y1)

x0 -= 5

y0 -= 5

x1 += 5

y1 += 5

canvas.pack()

root.mainloop()

```

#### 实例059：画图（丑）

\*\*题目：\*\*画图，综合例子。

\*\*程序分析：\*\*丑。

```

if \_\_name\_\_ == '\_\_main\_\_':

from tkinter import \*

canvas = Canvas(width = 300,height = 300,bg = 'green')

canvas.pack(expand = YES,fill = BOTH)

x0 = 150

y0 = 100

canvas.create\_oval(x0 - 10,y0 - 10,x0 + 10,y0 + 10)

canvas.create\_oval(x0 - 20,y0 - 20,x0 + 20,y0 + 20)

canvas.create\_oval(x0 - 50,y0 - 50,x0 + 50,y0 + 50)

import math

B = 0.809

for i in range(16):

a = 2 \* math.pi / 16 \* i

x = math.ceil(x0 + 48 \* math.cos(a))

y = math.ceil(y0 + 48 \* math.sin(a) \* B)

canvas.create\_line(x0,y0,x,y,fill = 'red')

canvas.create\_oval(x0 - 60,y0 - 60,x0 + 60,y0 + 60)

for k in range(501):

for i in range(17):

a = (2 \* math.pi / 16) \* i + (2 \* math.pi / 180) \* k

x = math.ceil(x0 + 48 \* math.cos(a))

y = math.ceil(y0 + 48 + math.sin(a) \* B)

canvas.create\_line(x0,y0,x,y,fill = 'red')

for j in range(51):

a = (2 \* math.pi / 16) \* i + (2\* math.pi / 180) \* k - 1

x = math.ceil(x0 + 48 \* math.cos(a))

y = math.ceil(y0 + 48 \* math.sin(a) \* B)

canvas.create\_line(x0,y0,x,y,fill = 'red')

mainloop()

```

#### 实例060：字符串长度

\*\*题目：\*\*计算字符串长度。

\*\*程序分析：\*\*无。

```

s='zhangguang101'

print(len(s))

```

#### 实例061：杨辉三角

\*\*题目：\*\*打印出杨辉三角形前十行。

\*\*程序分析：\*\*无。

```

def generate(numRows):

r = [[1]]

for i in range(1,numRows):

r.append(list(map(lambda x,y:x+y, [0]+r[-1],r[-1]+[0])))

return r[:numRows]

a=generate(10)

for i in a:

print(i)

```

#### 实例062：查找字符串

\*\*题目：\*\*查找字符串。

\*\*程序分析：\*\*无。

```

s1='aabbxuebixuebi'

s2='ab'

s3='xue'

print(s1.find(s2))

print(s1.find(s3))

```

#### 实例063：画椭圆

\*\*题目：\*\*画椭圆。

\*\*程序分析：\*\*使用 tkinter。

```

if \_\_name\_\_ == '\_\_main\_\_':

from tkinter import \*

x = 360

y = 160

top = y - 30

bottom = y - 30

canvas = Canvas(width = 400,height = 600,bg = 'white')

for i in range(20):

canvas.create\_oval(250 - top,250 - bottom,250 + top,250 + bottom)

top -= 5

bottom += 5

canvas.pack()

mainloop()

```

#### 实例64：画椭圆、矩形

\*\*题目：\*\*利用ellipse 和 rectangle 画图。。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

from tkinter import \*

canvas = Canvas(width = 400,height = 600,bg = 'white')

left = 20

right = 50

top = 50

num = 15

for i in range(num):

canvas.create\_oval(250 - right,250 - left,250 + right,250 + left)

canvas.create\_oval(250 - 20,250 - top,250 + 20,250 + top)

canvas.create\_rectangle(20 - 2 \* i,20 - 2 \* i,10 \* (i + 2),10 \* ( i + 2))

right += 5

left += 5

top += 10

canvas.pack()

mainloop()

```

#### 实例065：画组合图形

\*\*题目：\*\*一个最优美的图案。

\*\*程序分析：\*\*无。

```

import math

from tkinter import \*

class PTS:

def \_\_init\_\_(self):

self.x = 0

self.y = 0

points = []

def LineToDemo():

screenx = 400

screeny = 400

canvas = Canvas(width = screenx,height = screeny,bg = 'white')

AspectRatio = 0.85

MAXPTS = 15

h = screeny

w = screenx

xcenter = w / 2

ycenter = h / 2

radius = (h - 30) / (AspectRatio \* 2) - 20

step = 360 / MAXPTS

angle = 0.0

for i in range(MAXPTS):

rads = angle \* math.pi / 180.0

p = PTS()

p.x = xcenter + int(math.cos(rads) \* radius)

p.y = ycenter - int(math.sin(rads) \* radius \* AspectRatio)

angle += step

points.append(p)

canvas.create\_oval(xcenter - radius,ycenter - radius,

xcenter + radius,ycenter + radius)

for i in range(MAXPTS):

for j in range(i,MAXPTS):

canvas.create\_line(points[i].x,points[i].y,points[j].x,points[j].y)

canvas.pack()

mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

LineToDemo()

```

#### 实例066：三数排序

\*\*题目：\*\*输入3个数a,b,c，按大小顺序输出。

\*\*程序分析：\*\*同实例005。

```

raw=[]

for i in range(3):

x=int(input('int%d: '%(i)))

raw.append(x)

for i in range(len(raw)):

for j in range(i,len(raw)):

if raw[i]>raw[j]:

raw[i],raw[j]=raw[j],raw[i]

print(raw)

raw2=[]

for i in range(3):

x=int(input('int%d: '%(i)))

raw2.append(x)

print(sorted(raw2))

```

#### 实例067：交换位置

\*\*题目：\*\*输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

\*\*程序分析：\*\*无。

```

li=[3,2,5,7,8,1,5]

li[-1],li[li.index(min(li))]=li[li.index(min(li))],li[-1]

m=li[0]

ind=li.index(max(li))

li[0]=li[ind]

li[ind]=m

print(li)

```

#### 实例068：旋转数列

\*\*题目：\*\*有n个整数，使其前面各数顺序向后移m个位置，最后m个数变成最前面的m个数

\*\*程序分析：\*\*无。

```

from collections import \*

li=[1,2,3,4,5,6,7,8,9]

deq=deque(li,maxlen=len(li))

print(li)

deq.rotate(int(input('rotate:')))

print(list(deq))

```

#### 实例069：报数

\*\*题目：\*\*有n个人围成一圈，顺序排号。从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的那位。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

nmax = 50

n = int(input('请输入总人数:'))

num = []

for i in range(n):

num.append(i + 1)

i = 0

k = 0

m = 0

while m < n - 1:

if num[i] != 0 : k += 1

if k == 3:

num[i] = 0

k = 0

m += 1

i += 1

if i == n : i = 0

i = 0

while num[i] == 0: i += 1

print(num[i])

```

#### 实例070：字符串长度II

\*\*题目：\*\*写一个函数，求一个字符串的长度，在main函数中输入字符串，并输出其长度。

\*\*程序分析：\*\*无。

```

def lenofstr(s):

return len(s)

print(lenofstr('tanxiaofengsheng'))

```

#### 实例071：输入和输出

\*\*题目：\*\*编写input()和output()函数输入，输出5个学生的数据记录。

\*\*程序分析：\*\*无。

```

N = 3

#stu

# num : string

# name : string

# score[4]: list

student = []

for i in range(5):

student.append(['','',[]])

def input\_stu(stu):

for i in range(N):

stu[i][0] = input('input student num:\n')

stu[i][1] = input('input student name:\n')

for j in range(3):

stu[i][2].append(int(input('score:\n')))

def output\_stu(stu):

for i in range(N):

print ('%-6s%-10s' % ( stu[i][0],stu[i][1] ))

for j in range(3):

print ('%-8d' % stu[i][2][j])

if \_\_name\_\_ == '\_\_main\_\_':

input\_stu(student)

print (student)

output\_stu(student)

```

#### 实例072：创建链表

\*\*题目：\*\*创建一个链表。

\*\*程序分析：\*\*原文不太靠谱。

```

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

def get\_data(self):

return self.data

class List:

def \_\_init\_\_(self, head):

self.head = head

def is\_empty(self):

return self.get\_len() == 0

def get\_len(self):

length = 0

temp = self.head

while temp is not None:

length += 1

temp = temp.next

return length

def append(self, node):

temp = self.head

while temp.next is not None:

temp = temp.next

temp.next = node

def delete(self, index):

if index < 1 or index > self.get\_len():

print("给定位置不合理")

return

if index == 1:

self.head = self.head.next

return

temp = self.head

cur\_pos = 0

while temp is not None:

cur\_pos += 1

if cur\_pos == index-1:

temp.next = temp.next.next

temp = temp.next

def insert(self, pos, node):

if pos < 1 or pos > self.get\_len():

print("插入结点位置不合理")

return

temp = self.head

cur\_pos = 0

while temp is not Node:

cur\_pos += 1

if cur\_pos == pos-1:

node.next = temp.next

temp.next =node

break

temp = temp.next

def reverse(self, head):

if head is None and head.next is None:

return head

pre = head

cur = head.next

while cur is not None:

temp = cur.next

cur.next = pre

pre = cur

cur = temp

head.next = None

return pre

def print\_list(self, head):

init\_data = []

while head is not None:

init\_data.append(head.get\_data())

head = head.next

return init\_data

if \_\_name\_\_=='\_\_main\_\_':

head=Node('head')

link=List(head)

for i in range(10):

node=Node(i)

link.append(node)

print(link.print\_list(head))

```

#### 实例073：反向输出链表

\*\*题目：\*\*反向输出一个链表。

\*\*程序分析：\*\*无。

```

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

def get\_data(self):

return self.data

class List:

def \_\_init\_\_(self, head):

self.head = head

def is\_empty(self):

return self.get\_len() == 0

def get\_len(self):

length = 0

temp = self.head

while temp is not None:

length += 1

temp = temp.next

return length

def append(self, node):

temp = self.head

while temp.next is not None:

temp = temp.next

temp.next = node

def delete(self, index):

if index < 1 or index > self.get\_len():

print("给定位置不合理")

return

if index == 1:

self.head = self.head.next

return

temp = self.head

cur\_pos = 0

while temp is not None:

cur\_pos += 1

if cur\_pos == index-1:

temp.next = temp.next.next

temp = temp.next

def insert(self, pos, node):

if pos < 1 or pos > self.get\_len():

print("插入结点位置不合理")

return

temp = self.head

cur\_pos = 0

while temp is not Node:

cur\_pos += 1

if cur\_pos == pos-1:

node.next = temp.next

temp.next =node

break

temp = temp.next

def reverse(self, head):

if head is None and head.next is None:

return head

pre = head

cur = head.next

while cur is not None:

temp = cur.next

cur.next = pre

pre = cur

cur = temp

head.next = None

return pre

def print\_list(self, head):

init\_data = []

while head is not None:

init\_data.append(head.get\_data())

head = head.next

return init\_data

if \_\_name\_\_=='\_\_main\_\_':

head=Node('head')

link=List(head)

for i in range(10):

node=Node(i)

link.append(node)

print(link.print\_list(head))

print(link.print\_list(link.reverse(head)))

```

#### 实例074：列表排序、连接

\*\*题目：\*\*列表排序及连接。

\*\*程序分析：\*\*排序可使用 sort() 方法，连接可以使用 + 号或 extend() 方法。

```

a=[2,6,8]

b=[7,0,4]

a.extend(b)

a.sort()

print(a)

```

#### 实例075：不知所云

\*\*题目：\*\*放松一下，算一道简单的题目。

\*\*程序分析：\*\*鬼知道是什么。

```

if \_\_name\_\_ == '\_\_main\_\_':

for i in range(5):

n = 0

if i != 1: n += 1

if i == 3: n += 1

if i == 4: n += 1

if i != 4: n += 1

if n == 3: print (64 + i)

```

#### 实例076：做函数

\*\*题目：\*\*编写一个函数，输入n为偶数时，调用函数求1/2+1/4+...+1/n,当输入n为奇数时，调用函数1/1+1/3+...+1/n

\*\*程序分析：\*\*无。

```

def peven(n):

i = 0

s = 0.0

for i in range(2,n + 1,2):

s += 1.0 / i

return s

def podd(n):

s = 0.0

for i in range(1, n + 1,2):

s += 1.0 / i

return s

def dcall(fp,n):

s = fp(n)

return s

if \_\_name\_\_ == '\_\_main\_\_':

n = int(input('input a number: '))

if n % 2 == 0:

sum = dcall(peven,n)

else:

sum = dcall(podd,n)

print (sum)

```

#### 实例077：遍历列表

\*\*题目：\*\*循环输出列表

\*\*程序分析：\*\*无。

```

l=['moyu','niupi','xuecaibichi','shengfaji','42']

for i in range(len(l)):

print(l[i])

```

#### 实例078：字典

\*\*题目：\*\*找到年龄最大的人，并输出。请找出程序中有什么问题。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

person = {"li":18,"wang":50,"zhang":20,"sun":22}

m = 'li'

for key in person.keys():

if person[m] < person[key]:

m = key

print ('%s,%d' % (m,person[m]))

```

#### 实例079：字符串排序

\*\*题目：\*\*字符串排序。

\*\*程序分析：\*\*无。

```

l=['baaa','aaab','aaba','aaaa','abaa']

l.sort()

print(l)

```

#### 实例080：猴子分桃

\*\*题目：\*\*海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子平均分为五份，多了一个，这只猴子把多的一个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，又多了一个，它同样把多的一个扔入海中，拿走了一份，第三、第四、第五只猴子都是这样做的，问海滩上原来最少有多少个桃子？

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

i = 0

j = 1

x = 0

while (i < 5) :

x = 4 \* j

for i in range(0,5) :

if(x%4 != 0) :

break

else :

i += 1

x = (x/4) \* 5 +1

j += 1

print(x)

for p in range(5):

x=(x-1)/5\*4

print(x)

```

#### 实例081：求未知数

\*\*题目：\*\*809\*??=800\*??+9\*?? 其中??代表的两位数, 809\*??为四位数，8\*??的结果为两位数，9\*??的结果为3位数。求??代表的两位数，及809\*??后的结果。

\*\*程序分析：\*\*无。

```

a = 809

for i in range(10,100):

b = i \* a

if b >= 1000 and b <= 10000 and 8 \* i < 100 and 9 \* i >= 100:

print(b,' = 800 \* ', i, ' + 9 \* ', i)

for i in range(10,100):

if 8\*i>99 or 9\*i<100:

continue

if 809\*i==800\*i+9\*i:

print(i)

break

```

#### 实例082：八进制转十进制

\*\*题目：\*\*八进制转换为十进制

\*\*程序分析：\*\*无。

```

n=eval('0o'+str(int(input('八进制输入：'))))

print(n)

```

#### 实例083：制作奇数

\*\*题目：\*\*求0—7所能组成的奇数个数。

\*\*程序分析：\*\*

组成1位数是4个。1,3,5,7结尾

组成2位数是7\*4个。第一位不能为0

组成3位数是7\*8\*4个。中间随意

组成4位数是7\*8\*8\*4个。

```

if \_\_name\_\_ == '\_\_main\_\_':

sum = 4

s = 4

for j in range(2,9):

print (sum)

if j <= 2:

s \*= 7

else:

s \*= 8

sum += s

print('sum = %d' % sum)

```

#### 实例084：连接字符串

\*\*题目：\*\*连接字符串。

\*\*程序分析：\*\*无。

```

delimiter = ','

mylist = ['Brazil', 'Russia', 'India', 'China']

print(delimiter.join(mylist))

```

#### 实例085：整除

\*\*题目：\*\*输入一个奇数，然后判断最少几个 9 除于该数的结果为整数。

\*\*程序分析：\*\*999999 / 13 = 76923。

```

if \_\_name\_\_ == '\_\_main\_\_':

zi = int(input('输入一个数字:'))

n1 = 1

c9 = 1

m9 = 9

sum = 9

while n1 != 0:

if sum % zi == 0:

n1 = 0

else:

m9 \*= 10

sum += m9

c9 += 1

print ('%d 个 9 可以被 %d 整除 : %d' % (c9,zi,sum))

r = sum / zi

print ('%d / %d = %d' % (sum,zi,r))

```

#### 实例086：连接字符串II

\*\*题目：\*\*两个字符串连接程序。

\*\*程序分析：\*\*无。

```

a='guangtou'

b='feipang'

print(b+a)

```

#### 实例087：访问类成员

\*\*题目：\*\*回答结果（结构体变量传递）。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

class student:

x = 0

c = 0

def f(stu):

stu.x = 20

stu.c = 'c'

a= student()

a.x = 3

a.c = 'a'

f(a)

print(a.x,a.c)

```

#### 实例088：打印星号

\*\*题目：\*\*读取7个数（1—50）的整数值，每读取一个值，程序打印出该值个数的＊。

\*\*程序分析：\*\*无。

```

for i in range(3):

print('\*'\*int(input('input a number: ')))

```

#### 实例089：解码

\*\*题目：\*\*某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：每位数字都加上5,然后用和除以10的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换。

\*\*程序分析：\*\*无。

```

n=input()

n = str(n)

a=[]

for i in range(4):

a.append((int(n[i])+5)%10)

a[0],a[3]=a[3],a[0]

a[1],a[2]=a[2],a[1]

print ("".join('%s' %s for s in a))

```

#### 实例090：列表详解

\*\*题目：\*\*列表使用实例。

\*\*程序分析：\*\*无。

```

#list

#新建列表

testList=[10086,'中国移动',[1,2,4,5]]

#访问列表长度

print (len(testList) )

#到列表结尾

print (testList[1:])

#向列表添加元素

testList.append('i\'m new here!')

print (len(testList) )

print (testList[-1] )

#弹出列表的最后一个元素

print (testList.pop(1) )

print (len(testList) )

print (testList )

#list comprehension

#后面有介绍，暂时掠过

matrix = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

print (matrix )

print (matrix[1] )

col2 = [row[1] for row in matrix]#get a column from a matrix

print (col2 )

col2even = [row[1] for row in matrix if row[1] % 2 == 0]#filter odd item

print (col2even)

```

#### 实例091：time模块

\*\*题目：\*\*时间函数举例1。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

import time

print (time.ctime(time.time()))

print (time.asctime(time.localtime(time.time())))

print (time.asctime(time.gmtime(time.time())))

```

#### 实例092：time模块II

\*\*题目：\*\*时间函数举例2。

\*\*程序分析：\*\*如何浪费时间。

```

if \_\_name\_\_ == '\_\_main\_\_':

import time

start = time.time()

for i in range(3000):

print(i)

end = time.time()

print (end - start)

```

#### 实例093：time模块III

\*\*题目：\*\*时间函数举例3。

\*\*程序分析：\*\*如何浪费时间。

```

if \_\_name\_\_ == '\_\_main\_\_':

import time

start = time.clock()

for i in range(100):

print(i)

end = time.clock()

print('different is %6.3f' % (end - start))

```

#### 实例094：time模块IV

\*\*题目：\*\*时间函数举例4。

\*\*程序分析：\*\*如何浪费时间。

```

if \_\_name\_\_ == '\_\_main\_\_':

import time

import random

play\_it = input('do you want to play it.(\'y\' or \'n\')')

while play\_it == 'y':

c = input('input a character:\n')

i = random.randint(0,2\*\*32) % 100

print ('please input number you guess:\n')

start = time.clock()

a = time.time()

guess = int(input('input your guess:\n'))

while guess != i:

if guess > i:

print('please input a little smaller')

guess = int(input('input your guess:\n'))

else:

print('please input a little bigger')

guess = int(input('input your guess:\n'))

end = time.clock()

b = time.time()

var = (end - start) / 18.2

print (var)

# print 'It took you %6.3 seconds' % time.difftime(b,a))

if var < 15:

print ('you are very clever!')

elif var < 25:

print ('you are normal!')

else:

print ('you are stupid!')

print ('Congradulations')

print ('The number you guess is %d' % i)

play\_it = input('do you want to play it.')

```

#### 实例095：转换时间格式

\*\*题目：\*\*字符串日期转换为易读的日期格式。

\*\*程序分析：\*\*看看就得了，dateutil是个第三方库。

```

from dateutil import parser

dt = parser.parse("Aug 28 2015 12:00AM")

print (dt)

```

#### 实例096：计算复读次数

\*\*题目：\*\*计算字符串中子串出现的次数。

\*\*程序分析：\*\*无。

```

s1='xuebixuebixuebixuebixuebixuebixuebixue'

s2='xuebi'

print(s1.count(s2))

```

#### 实例097：磁盘写入

\*\*题目：\*\*从键盘输入一些字符，逐个把它们写到磁盘文件上，直到输入一个 # 为止。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

from sys import stdout

filename = input('输入文件名:\n')

fp = open(filename,"w")

ch = input('输入字符串:\n')

while ch != '#':

fp.write(ch)

stdout.write(ch)

ch = input('')

fp.close()

```

#### 实例098：磁盘写入II

\*\*题目：\*\*从键盘输入一个字符串，将小写字母全部转换成大写字母，然后输出到一个磁盘文件"test"中保存。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

fp = open('test.txt','w')

string = input('please input a string:\n')

string = string.upper()

fp.write(string)

fp = open('test.txt','r')

print (fp.read())

fp.close()

```

#### 实例099：磁盘读写

\*\*题目：\*\*有两个磁盘文件A和B,各存放一行字母,要求把这两个文件中的信息合并(按字母顺序排列), 输出到一个新文件C中。

\*\*程序分析：\*\*无。

```

if \_\_name\_\_ == '\_\_main\_\_':

import string

fp = open('test1.txt')

a = fp.read()

fp.close()

fp = open('test2.txt')

b = fp.read()

fp.close()

fp = open('test3.txt','w')

l = list(a + b)

l.sort()

s = ''

s = s.join(l)

fp.write(s)

fp.close()

```

#### 实例100：列表转字典

\*\*题目：\*\*列表转换为字典。

\*\*程序分析：\*\*无。

```

i = ['a', 'b']

l = [1, 2]

print (dict(zip(i,l)))

```