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

题目001：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？

'''

def tm001():

arr = []

for i in range(1,5):

for j in range(1,5):

for k in range(1,5):

num = 100\*i+10\*j+k

if i!=j and j!=k and i!=k and num not in arr:# 互不相同且无重复数字的三位数

arr.append(num)

print(len(arr),arr)

def tm001\_1():

'''

【个人备注】：其实python自带排列组合模块，可以直接调用。

也知道这个写法，只是函数记不住，还是百度一下才能写出来。

如果这是面试题，能写出后一种当然好，不能的话还是老老实实的按照上面的思路来吧。

'''

import itertools

temp\_arr = list(itertools.permutations([1, 2, 3, 4], 3)) # 排列 # A\_4^3 = (4)!/(4-3)! = (4\*3\*2\*1)/1 = 24

arr = [100\*t[0]+10\*t[1]+t[2] for t in temp\_arr]

print(len(arr),arr)

'''

题目002：企业发放的奖金根据利润(I)的多少来提成：

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？

'''

def tm002():

'''

程序分析：请利用数轴来分界，定位。

【个人备注】：这种处理数轴问题的写法，值得参考。比elif的写法，简洁方便的多。

'''

money = int(input('净利润:'))

arr = [1000000,600000,400000,200000,100000,0]

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

bonus = 0

for i in range(len(arr)):

if money>arr[i]: # 对于处于区间的部分

bonus+=(money-arr[i])\*rat[i] # 计算并累加奖励

money=arr[i] # 剩余部分

print(bonus)

'''

题目003：一个整数，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？

'''

def tm003():

'''

【个人备注】：网站上是求了一下方程，没细看。

python又不是没有开方函数，直接按字面意思解了。

'''

import math

for i in range(1000):

x = math.sqrt(i+100)

y = math.sqrt(i+100+168)

if x%1==0 and y%1==0:

print(i)

'''

题目004：输入某年某月某日，判断这一天是这一年的第几天？

'''

def tm004():

'''

【个人备注】：知道python有时间元组这一概念，这道题完全不需要计算。

时间元组包含九个属性

tm\_year 年

tm\_mon 月(1~12)

tm\_mday 日(1~31)

tm\_hour 时(0~23)

tm\_min 分(0~59)

tm\_sec 秒(0~61, 60或61是闰秒)

tm\_wday 星期(0~6, 0是周一)

tm\_yday 第几天(1~366, 366是儒略历)

tm\_isdst 夏令时(平时用不到)

'''

import time

date = input('输入时间(例如2018-01-23):')

st = time.strptime(date,'%Y-%m-%d') # 时间文本转化成时间元祖

num = st.tm\_yday

print(num)

'''

题目005：输入三个整数x,y,z，请把这三个数由小到大输出。

'''

def tm005():

print('输入三个数字')

x = input('输入第1个数字:')

y = input('输入第2个数字:')

z = input('输入第3个数字:')

l = [x,y,z]

arr = sorted(l) # 你也可以使用list.sort()方法来排序，此时list本身将被修改

print(arr)

'''

题目006：斐波那契数列。

'''

def tm006():

'''

程序分析：斐波那契数列，又称黄金分割数列，指的是这样一个数列：0、1、1、2、3、5、8、13、21、34、……。

【个人备注】：很多种解法，我是按照分割的方式，每次取列表后两项，然后相加。

'''

l = [0,1]

for i in range(10):

arr = l[-2:]

l.append(arr[0]+arr[1])

print(l)

'''

题目007：将一个列表的数据复制到另一个列表中。

'''

def tm007():

'''

【个人备注】：如果系统的看过python教程之类的应该都知道。

Python里面一切都是对象，list的复制需要用[:]的方式。

至于b=a只是相当于给a取了个别名而已，指向的是同一个列表，并没有实现复制。

'''

a = [1, 2, 3]

b = a[:]

'''题外话'''

a[0]=0

print(id(a),id(b)) # 可以看到a,b的内存不一致，是复制

print(a,b) # 修改a，b不变

a = [1, 2, 3]

b = a

a[0]=0

print(id(a),id(b)) # 如果去掉[:]，可以看到a,b的内存一致，并没有复制，指向的是同一个列表

print(a,b) # 修改a，b也变

'''

题目008：题目：输出 9\*9 乘法口诀表。

'''

def tm008():

#【个人备注】：已经忘了，百度了才想起来口诀表具体长什么样。

# 注意 %-7s 和 end='' 的用法，其他没什么。

for i in range(1,10):

for j in range(1,10):

if j<=i:

string = '%d\*%d=%d'%(j,i,j\*i)

print('%-7s'%string,end='')

print('')

'''

题目009：暂停一秒输出。

'''

def tm009():

#【个人备注】：time.sleep()，用过的都知道。

import time

a = time.time()

time.sleep(1)

b = time.time()

print(b-a)

'''

题目010：暂停一秒输出，并格式化当前时间。

'''

def tm010():

#【个人备注】：用过几次，用过就忘。

import time

a = time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time())) # time.localtime()时间戳转化成时间元祖

print(a)

time.sleep(1)

b = time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time())) # time.strftime()时间元祖转化成时间文本

print(b)

'''

题目011：古典问题：

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？

'''

def tm011():

'''

其实这道题就是斐波那契数列的由来。

【个人备注】：理清思路是关键，理解成满两个月后，每月都能生兔子，就好办了。

'''

m1=1 # 满月

m2=0 # 满俩月（下个月生兔子）

mm=0 # 可以月月生兔子了

for i in range(1,10):

# 过了1个月后

mm = mm+m2 # 加入新增成年的兔子

m2 = m1 # 满月的变成满俩月

m1 = mm # 这个月新出生兔子

print(i,mm+m1+m2) # 每个月有多少对兔子

'''

题目012：判断101-200之间有多少个素数，并输出所有素数。

'''

def tm012():

'''

【个人备注】：按照素数不能被之前的素数整除，取200以内所有素数，然后取出101-200之间的部分。

'''

arr = [2,3]

# 取200以内所有素数

for i in range(4,201):

for j in arr:

if i%j==0:

break

# else: # 这是一开始我自己的写法，后来发现for可以直接接else子语句

# if j==arr[-1]:

# arr.append(i)

else: # 迭代的对象成功迭代完，位于else的子句将执行；而如果在for循环中含有break时则直接终止循环，并不会执行else子句。

arr.append(i)

# 取出100-200之间部分

for i in range(len(arr)):

if arr[i]>100:

l = arr[i:]

print(len(l),l)

break

'''

题目013：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。

'''

def tm013():

'''

【个人备注】： // 取整，% 求余，\*\*3 三次方。知道取整求玉写法的就没问题。

'''

for i in range(100,1000):

b = i//100 # 百位

s = i%100//10 # 十位

g = i%10 # 个位

if b\*\*3+s\*\*3+g\*\*3==i:

print(i)

'''

题目014：将一个正整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5。

'''

def tm014():

'''

【个人备注】：拆到拆不动为止，类似012题。

'''

import math

num = int(input('输入一个整数:'))

arr = []

while num>1:

for i in range(2,int(math.sqrt(num))+1): # 因为题目是一个没写范围正整数，开方可以有效减少该值过大时候的计算量

if num%i==0:

arr.append(i)

num = num//i

break

else:

arr.append(num)

break

print(arr)

'''

题目015：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。

'''

def tm015():

'''

【个人备注】：if-else基本用法，没啥说的。

'''

score = float(input('输入一个成绩:'))

if score>=90:

print('A')

elif score>=60:

print('B')

else:

print('C')

'''

题目016：输出指定格式的日期。

'''

def tm016():

'''

【个人备注】：用的不多经常忘，整理了一下参考答案和一些转换

'''

import time

print(time.time()) # 时间戳 1498539133.655

print(time.localtime()) # 时间元祖 tm\_year=2017, tm\_mon=6, tm\_mday=27, tm\_hour=12, tm\_min=53, tm\_sec=16, tm\_wday=1, tm\_yday=178, tm\_isdst=0

print(time.asctime()) # 时间的一种可读文本形式 'Tue Jun 27 12:53:50 2017'

print(time.strftime('%Y-%m-%d %H:%M:%S',time.localtime())) # 按指定文本格式输出时间 '2017-06-27 13:00:57'

st = time.localtime(time.time()) # 时间戳 转化成 时间元祖

st = time.strptime('2018/1/23','%Y/%m/%d') # 时间文本 转化成 时间元祖

date = time.strftime('%Y-%m-%d',st) # 时间元祖 转化成 时间文本 '%Y-%m-%d %H:%M:%S'

print(date) # 前面两条函数配合着用，相当于将时间文本重新格式化。

# 另外我们可以通过datetime模块来计算时间差，例如：

import datetime

dt1 = datetime.datetime.fromtimestamp(1517302458)

print(dt1,type(dt1))

dt2 = datetime.datetime.now()

print(dt2)

print('相差%d天零%.1f个小时'%((dt2-dt1).days,(dt2-dt1).seconds/60/60))

'''

2018-01-30 16:54:18 <class 'datetime.datetime'>

2018-02-01 16:27:47.524774

相差1天零23.6个小时

'''

# 注意上面的日期虽然看起来是文本，但实际上是datetime类型的。

# 可以通过时间戳/时间文本转换得到，然后才能进行日期时间计算。

d1 = datetime.datetime.strptime('2017-10-16 19:21:22', '%Y-%m-%d %H:%M:%S')

'''

题目017：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。

'''

def tm017():

'''

【个人备注】：本来想写成这种格式来着

import string

if c in string.ascii\_letters: # abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

if c == ' ': # 空格

if c in string.digits: # 0123456789

看了参考答案才知道有现成的函数可以用。

其实随便哪种都行，甚至直接把一串字母粘上去看起来更直接。

'''

s = raw\_input('input a string:\n')

letters,space,digit,others = 0,0,0,0

for c in s:

if c.isalpha():

letters += 1

elif c.isspace():

space += 1

elif c.isdigit():

digit += 1

else:

others += 1

print('char = %d,space = %d,digit = %d,others = %d' % (letters,space,digit,others))

'''

题目018：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。

'''

def tm018():

'''

【个人备注】：答案给的解法很多种，但是我还是认为我写的方法最简单。

2+22+222+2222+22222

可以理解为：

20000 + 2\*2000 + 3\*200 + 4\*20 + 5\*2

也就是：

1\*2\*10^4 + 2\*2\*10^3 + 3\*2\*10^2 + 4\*2\*10^1 + 5\*2\*10^0

所以简单迭代就可以出结果

'''

a = 2

t = 5

num = 0

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

num+=i\*a\*(10\*\*(t-i))

print(num)

'''

题目019：一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。

'''

def tm019():

'''

【个人备注】：题意看的不是太懂，于是百度了一下：完数就是除了自身之外的所有约数之和等于他本身。

第一个完全数是6，它有约数1、2、3、6，除去它本身6外，其余3个数相加，1+2+3=6。

第二个完全数是28，它有约数1、2、4、7、14、28，除去它本身28外，其余5个数相加，1+2+4+7+14=28。

终于看懂了题意。

先求出所有约数，然后求和比一下是否相等就行了，没有难度

'''

for num in range(1,1000):

arr = []

for i in range(1,num):

if num%i==0:

arr.append(i)

if sum(arr)==num:

print(num,arr)

'''

题目020：一球从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？

'''

def tm020():

'''

【个人备注】：很简单，没啥说的

'''

total = 0

m = 100 # 第一次落地，经过了一百米

total += m

for i in range(10-1): # 之后9次弹起到落地

m = m/2 # 弹起的高度

total += 2\*m # 弹起然后重新落地，一共经过的距离

print(total)

print(m/2)

'''

题目021：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。

'''

def tm021():

'''

【个人备注】：第十天num=1个，第九天必然是4个：4/2-1=1，也就是(num+1)\*2=4。

做这种题，先用算数式列出来，然后用代码描述就行了。

'''

num = 1

for i in range(10-1):

num = (num+1)\*2

print(num)

'''

题目022：两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。

'''

def tm022():

'''

【个人备注】：关键是将抽象化，将问题抽象成代码方式。

我的解题思路，是用排列组合函数，列出方案，然后排除。

而官方解答里面有一个纯粹的for循环加if的求解方式，

更抽象一些，用了一个很常用固定范式，直接拿来解题了。

稍微对其抽象的方法注释了一下。

'''

import itertools

jia = ['a','b','c']

yi = ['x','y','z']

arr = list(itertools.permutations(yi,3)) # 面对甲队a,b,c时，乙队所有排列 [('x', 'y', 'z'), ('x', 'z', 'y'), ('y', 'x', 'z'), ('y', 'z', 'x'), ('z', 'x', 'y'), ('z', 'y', 'x')]

arr = [[jia[i]+a[i] for i in range(3)] for a in arr] #将a,b,c写上，得到所有对阵组合 [['ax', 'by', 'cz'], ['ax', 'bz', 'cy'], ['ay', 'bx', 'cz'], ['ay', 'bz', 'cx'], ['az', 'bx', 'cy'], ['az', 'by', 'cx']]

for i in arr:

if 'ax' in i:

pass

elif 'cx' in i or 'cz' in i:

pass

else:

print(i) # 得到 ['az', 'bx', 'cy']

def tm022\_1():

for a in ['x','y','z']: # a在x,y,z中挑一个打

for b in ['x', 'y', 'z']: # b在x,y,z中挑一个打

for c in ['x', 'y', 'z']: # c在x,y,z中挑一个打 三层总计27种打法

if a!=b and b!=c and c!=a: # a,b,c不能挑同一个人 缩减到6种打法

if a!='x' and c!='x' and c!='z': # 依据题意不能ax,cx,cz 缩减到1种打法

print('a'+a,'b'+b,'c'+c)

'''

题目023：

打印出如下图案（菱形）:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

'''

def tm023():

'''

【个人备注】：想到了绝对值，

然后将[0,1,2,3,4,5,6]变成了[3,2,1,0,1,2,3]，也就是每行左边空格数。

'''

num = 7

for i in range(num):

blank = abs(num//2-i)

print(' '\*blank+'\*'\*(num-2\*blank)+' '\*blank)

'''

题目024：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。

'''

def tm024():

# 没啥好说的

a,b,num = 2,1,0

for i in range(20):

num+=a/b

a=a+b

b=a-b

print(num)

'''

题目025：求1+2!+3!+...+20!的和。

'''

def tm025():

'''

【个人备注】：实现起来很简单。

下面官方的代码，比我写的更简洁。

'''

s,t=0,1

for n in range(1,21):

t\*=n

s+=t

print(s)

'''

题目026：利用递归方法求5!。

'''

def fac(x):

if x>1:

return x\*fac(x-1)

else:

return x

def tm026():

'''

【个人备注】：按题目要求，公式f(n)=n\*f(n-1)，递归调用求解。

'''

print(fac(5))

'''

题目027：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。

'''

def output(s,l):

if l==0:

return

print (s[l-1])

output(s,l-1)

def tm027():

'''

【个人备注】：直接从官网复制，不喜欢递归。

'''

s = input('Input a string:')

l = len(s)

output(s,l)

'''

题目028：有5个人坐在一起，

问第五个人多少岁？他说比第4个人大两岁。

问第4个人，他说比第3个人大两岁。

问第3个人，又说比第2个人大两岁。

问第2个人，说比第1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？

'''

def age(x):

if x>1:

return 2+age(x-1)

else:

return 10

def tm028():

'''

【个人备注】：官网给的还是递归方法，因为不用递归的话就是个口算题。

'''

print(age(5))

'''

题目029：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。

'''

def tm029():

'''

【个人备注】：用Python，So Easy~

list倒序可以用list.reverse()；

字符串就只能用步长=-1的方式来倒序了。

'''

num = 12345

s = str(num)

print(len(s))

print(s[::-1])

'''

题目030：一个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。

'''

def tm030():

'''

【个人备注】：没啥可说的。

'''

num = 12321

s = str(num)

for i in range(len(s)//2):

if s[i]!=s[-i-1]:

print(False)

break

else:

print(True)

'''

题目031：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。

'''

def tm031():

'''

【个人备注】：按照题意要求实现了就行

'''

week = ['monday','tuesday','wednesday','thursday','friday','saturday','sunday']

inp = ''

while 1:

arr = []

inp = inp+input('请输入一个字母:')

for day in week: # 挑出满足输入的星期

if inp==day[:len(inp)]:

arr.append(day)

if len(arr)==1: # 只剩一个，说明唯一，可以输出结果

print('以%s开头的单词是:%s'%(inp,arr[0]))

inp=''

elif len(arr)==0: # 一个都没有说明输错了，需要重新输入

print('没有%s开头的单词'%inp)

inp=''

'''

题目032：按相反的顺序输出列表的值。

'''

def tm032():

'''

【个人备注】：之前已经做过了

'''

# 方法一

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

print(a[::-1])

# 方法二

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

a.reverse()

print(a)

# 方法三

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

a.sort(reverse=True)

print(a)

'''

题目033：按逗号分隔列表。

'''

def tm033():

'''

【个人备注】：一开始没看懂题目项干啥，直接print(list)不就是逗号分隔么

实际上题目的意思是，合并列表项，以逗号为间隔

'''

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

print(','.join(a))

'''

题目034：练习函数调用。

'''

def tm034():

'''

【个人备注】：没头没尾的一个题，之前的递归实际上就是函数的调用了。不写了。

'''

pass

'''

题目035：文本颜色设置。

'''

def tm035():

'''

【个人备注】：感觉不像是不是python的题。

'''

pass

'''

题目036：求100之内的素数。

'''

def tm036():

'''

【个人备注】：素数就是质数

'''

arr = [2]

for i in range(3,100):

for j in arr:

if i%j==0:

break

else:

arr.append(i)

print(arr)

'''

题目037：对10个数进行排序。

'''

def tm037():

'''

【个人备注】：实际上考察的是排序。揣测了一下题意，写了两种解法

'''

# 方法1，python解法

a = [1,5,7,3,2,4,9,10,6,8]

a.sort()

print(a)

# 方法2，常规解法

a = [1,5,7,3,2,4,9,10,6,8]

b = [a[0]]

for num in a[1:]:

for i in range(len(b)):

if num<b[i]:

b.insert(i,num)

break

else:

b.append(num)

print(b)

'''

题目038：求一个3\*3矩阵主对角线元素之和。

'''

def tm038():

'''

【个人备注】：思路比较简单，n\*n的矩阵都可以用这个求解。

'''

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

s = 0

n = len(a)

for i in range(n): # 左上到右下一条线

s+=a[i][i]

for i in range(n): # 右上到左下一条线

s+=a[i][n-i-1]

if n%2==1: # 如果是奇数，删掉重复计算的中间点

s-=a[n//2][n//2]

print(s)

'''

题目039：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。

'''

def tm039():

'''

【个人备注】：037题中刚写过。

'''

aaa = [1,5,8,14,28,39,60,89,134,324,612,900]

b = 555

for a in aaa:

if b<a:

aaa.insert(aaa.index(a),b)

break

else:

aaa.append(b)

print(aaa)

'''

题目040：将一个数组逆序输出。

'''

def tm040():

'''

【个人备注】：就是032题，重复了。

'''

pass

'''

题目041：模仿静态变量的用法。

'''

def tm041():

'''

【个人备注】：不是很清楚什么意思，直接看了官网的解答。

如果是函数中的局部变量，每次调用函数都会初始化。

而类中的变量，创建类的时候初始化，每次执行类中的函数的时候，不会初始化类变量。

看起来是想说这么个意思。

'''

def varfunc():

var = 0

print('var = %d' % var)

var += 1

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

for i in range(3):

varfunc()

# StaticVar作为类的一个属性，相当于静态变量

class Static:

StaticVar = 5

def varfunc(self):

self.StaticVar += 1

print(self.StaticVar)

print(Static.StaticVar)

a = Static()

for i in range(3):

a.varfunc()

'''

题目042：学习使用auto定义变量的用法。

'''

def tm042():

'''

【个人备注】：同样没看懂题意，看了下官网答案

发现实际是想说变量作用域， python是有分局部变量、全局变量的等区分的。

'''

num = 2

def autofunc():

num = 1

print('internal block num = %d'%num)

num += 1

for i in range(3):

print('The num = %d'%num)

num += 1

autofunc()

'''

以上实例输出结果为：

The num = 2

internal block num = 1

The num = 3

internal block num = 1

The num = 4

internal block num = 1

'''

'''

题目043：模仿静态变量(static)另一案例。

'''

def tm043():

'''

【个人备注】：官网的答案和041没啥区别，又来一遍

'''

'''

题目044：两个3\*3的矩阵，实现其对应位置的数据相加，并返回一个新矩阵：

X = [[12,7,3],

[ 4,5,6],

[ 7,8,9]]

Y = [[5,8,1],

[ 6,7,3],

[ 4,5,9]]

'''

def tm044():

'''

【个人备注】：自己写了一个，但是印象里python是有矩阵计算工具的。

官网就有人用该工具numpy写了一下，很简洁。

'''

x = [[12,7,3],[4,5,6],[7,8,9]]

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

z = x[:]

for i in range(3):

for j in range(3):

z[i][j]=x[i][j]+y[i][j]

print(z)

def tm044\_1():

import numpy # pip install numpy 需要安装模块

x = numpy.array([[12,7,3],[4,5,6],[7,8,9]])

y = numpy.array([[5,8,1],[6,7,3],[4,5,9]])

z = x+y

print(z)

'''

题目045：统计 1 到 100 之和。

'''

def tm045():

'''

【个人备注】：简单，但官网有人写的更简单

'''

s = 0

for i in range(1,101):

s+=i

print(s)

# 更简洁的方法

print(sum(range(1,101)))

'''

题目046：求输入数字的平方，如果平方运算后小于 50 则退出。

'''

def tm046():

'''

【个人备注】：简单

'''

while 1:

x= input('输入数字得到平方值:')

print(x\*x)

if x\*x<50:

break

'''

题目047：两个变量值互换。

'''

def tm047():

'''

【个人备注】：很简单

'''

a,b=1,2

a,b=b,a

print(a,b)

'''

题目048：数字比较。

'''

def tm048():

'''

【个人备注】：看了一眼官网答案，简单到懒得写。以下官网答案。

'''

i = 10

j = 20

if i > j:

print('%d 大于 %d' % (i,j))

elif i == j:

print('%d 等于 %d' % (i,j))

elif i < j:

print('%d 小于 %d' % (i,j))

else:

print('未知')

'''

题目049：使用lambda来创建匿名函数。

'''

def tm049():

'''

【个人备注】：用的不多又忘了，看了一下网上的教程，整理了一下。

'''

#lambda函数也叫匿名函数，即，函数没有具体的名称。先来看一个最简单例子：

def f(x):

return x\*\*2

print(f(4))

#Python中使用lambda的话，写成这样

g = lambda x:x\*\*2

print(g(4))

#lambda存在意义就是对简单函数的简洁表示。

#lambda语句中，冒号前是参数，可以有多个，用逗号隔开，冒号右边的返回值。

#常搭配内置函数map、filter、reduce，都是应用于序列的内置函数。常见的序列包括list、tuple、str。

#map(func, \*iterables) --> map object

#filter(function or None, iterable) --> filter object

#reduce(function, sequence[, initial]) -> value

foo = [2, 18, 9, 22, 17, 24, 8, 12, 27]

print(list(map(lambda x: x \* 2 + 10, foo))) # 映射 [14, 46, 28, 54, 44, 58, 26, 34, 64]

print(list(filter(lambda x: x % 3 == 0, foo))) # 过滤 [18, 9, 24, 12, 27]

from functools import reduce # 在Python 3里,reduce()函数已经被从全局名字空间里移除了,它现在被放置在fucntools模块里

print(reduce(lambda x, y: x + y, foo)) # 累积 139

'''

题目050：输出一个随机数。

'''

def tm050():

'''

【个人备注】：之前学习随机的时候整理的东西，用到时候来找就行了

'''

import random

# 随机数操作

random.random() # 0.85415370477785668 # 随机一个[0,1)之间的浮点数

random.uniform(0, 100) # 18.7356606526 # 随机一个[0,100]之间的浮点数

random.randrange(0, 100, 2) # 44 # 随机一个[0,100)之间的偶数

random.randint(0, 100) # 22 # 随机一个[0,100]之间的整数

# 随机字符操作

seed = "1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&\*()\_+=-" # 任意字符串（作为随机字符种子库）

random.choice(seed) # 'd' # 随机一个字符

random.sample(seed, 3) # ['a', 'd', 'b'] # 随机多个字符（字符可重复）

''.join(random.sample(seed,3)) # '^f^' # 随机指定长度字符串（字符可重复）

# 随机列表操作

random.shuffle(list) # 列表中的元素打乱

'''

题目051~053、055：

学习使用按位与 &

学习使用按位或 |

学习使用按位异或 ^

学习使用按位取反 ~

'''

def tm051(): # tm052 # tm053 # tm055

'''

【个人备注】：大概看了一下资料，按位操作可以通过bin()函数转换为二进制从而实现。

感觉二进制的计算如果不是学加密之类的基本用不上。

所以这一题，大概写了集合的用法，&、|在python中是集合set()才用的符号

两个列表求交集并集，通常也都是转换为集合，然后进行计算的。

'''

# 可迭代变量转换为集合形式

x = set('runoob')

y = set('google')

print(x, y) # 重复的被删除 {'n', 'o', 'b', 'u', 'r'} {'o', 'g', 'e', 'l'}

# 集合的交集、并集、差集

print(x & y) # 交集 {'o'}

print(x | y) # 并集 {'e', 'o', 'g', 'l', 'u', 'n', 'b', 'r'}

print(x - y) # 差集 {'n', 'b', 'u', 'r'}

# 当然也可以写成函数形式，不过确实没有上面符号好记。

print(x.intersection(y))

print(x.union(y))

print(x.difference(y))

'''

题目054：取一个整数a从右端开始的4-7位。

'''

def tm054():

'''

【个人备注】：官网答案不对

'''

a = 123456789

b = str(a)

print(b[-7:-3]) # 写的时候注意一下python切片是[-7:-4)左闭右开的，不包含[-4]的，所以要写成-3才能取到

'''

题目056：画图，学用circle画圆形。

题目057：画图，学用line画直线。

题目058：画图，学用rectangle画方形。

题目059：画图，综合例子。

题目063：画椭圆。

题目064：利用ellipse 和 rectangle 画图。

题目065：一个最优美的图案。

所有画图题pass 有爱的自己参照下面，将官网答案改成python3运行即可。

'''

def tm056(): # tm057、tm058、tm059、tm063、tm064、tm065、

'''

【个人备注】：这个技能感觉用不上啊。

复制了官网Python2答案，调整一下放到python3上成功运行了。

'''

import tkinter

canvas = tkinter.Canvas(width=600, height=500, bg='yellow')

canvas.pack(expand='yes', fill='both')

k = 1

j = 1

for i in range(0,26):

canvas.create\_oval(300 - k,250 - k,300 + k,250 + k, width=1)

k += j

j += 0.6

canvas.mainloop()

'''

题目060：计算字符串长度。

'''

def tm060():

'''

【个人备注】：无

'''

a = 'aegweg'

print(len(a))

'''

题目061：打印出杨辉三角形（要求打印出10行如下图）。

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

'''

def tm061():

'''

【个人备注】：感觉自己写的比官网简单好懂

'''

arr = [1]

print(1)

while len(arr)<10: # [1,1]

a = [0]+arr # [0,1,1]

b = arr+[0] # [1,1,0]

arr = [a[i]+b[i] for i in range(len(a))] # [1,2,1]

s = [str(i) for i in arr]

print(' '.join(s))

'''

题目062：查找字符串。

'''

def tm062():

s = 'abcde'

print(s.find('c'))

'''

题目066：输入3个数a,b,c，按大小顺序输出。

'''

def tm066():

arr=[]

for i in range(3):

a = input('请输入数字:')

arr.append(int(a))

arr.sort(reverse=True)

print('从大到小',arr)

'''

题目067：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。

'''

def tm067():

'''

【个人备注】：感觉自己写的比官网简单好懂，直接用python思维写，确实很方便

'''

a = [6,3,10,2,5,1,4,7,9,8]

i = a.index(max(a))

a[0],a[i] = a[i],a[0]

i = a.index(min(a))

a[-1],a[i] = a[i],a[-1]

print(a)

'''

题目068：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

'''

def tm068():

a = [1,2,3,4,5,6,7,8,9,10]

m = 3

b = a[-m:]+a[:-m]

print(b)

'''

题目069：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。

'''

def tm069():

'''

【个人备注】：python基础变量没有这种圈状循环的。

最直接的方法是列表，点队尾再从头开始。

缺点是列表删除中间项，后面的索引值都会变，需要计算。

所以我换了种方法，

把问题抽象成排队进门的问题

完全不考虑索引问题。

'''

# 初始化

n = 34

arr = list(range(1,n+1)) # 所有人门外站成一队【a】,进门后依旧按序站好【b】

count,a,b = 0,arr,[]

# 开始解题

while len(a+b)>1: # 循环直到只剩1人

num,count=a.pop(0),count+1 # 排队进门，每进一人【a.pop】，按一下计数器

if count%3!=0:b.append(num) # 进门后依旧按序站好【b.append】，计数器逢3淘汰。

if a==[]:a,b=b,[] # 如果门外没人了【a=[]】，所有人重新到门外站好【a=b】

print(a[0])

'''

题目070：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。

'''

def tm070():

'''

【个人备注】：简单

'''

def getlength(string):

return len(string)

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

x = 'abcde'

print(getlength(x))

'''

题目071：编写input()和output()函数输入，输出5个学生的数据记录。

'''

def tm071():

'''

【个人备注】：用字典类型随便写写

'''

def inp(data):

name = input('输入学生姓名：')

score = input('输入学生成绩：')

data[name]=score

print('成功录入')

return data

def outp(data):

name = input('输入学生姓名：')

print('该学生的成绩是：',data.get(name))

return data

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

data = {}

while 1:

a = input('输入/输出学生成绩(i/o)：')

if a=='i':

data = inp(data)

elif a=='o':

data = outp(data)

else:

print('输入值不对')

'''

题目072：创建一个链表。

题目073：反向输出一个链表。

'''

def tm072():

'''

【个人备注】：已经几乎忘了链表了，网上搜了一下终于想起链表是什么来着。

这种底层的东西，完全没必要用python去模拟

http://www.newsmth.net/nForum/#!article/Python/73818?p=10

'''

pass

'''

题目074：列表排序及连接。

题目079：字符串排序。(也是一样的)

'''

def tm074(): # tm079

'''

【个人备注】：之前都用过

'''

a = [3,2,1]

b = [4,5,6]

a.sort()

print(a)

print(a+b)

'''

题目075：放松一下，算一道简单的题目。

'''

def tm075():

'''

【个人备注】：Σ( ° △ °|||)︴，不是我偷工减料，题目就是这个，后面太监了

'''

pass

'''

题目076：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

'''

def tm076():

'''

【个人备注】：学了lambda想耍一下，结果发现官网写的比我还简洁！

'''

n =17

fenmu = range(2,n+1,2) if n%2==0 else range(1,n+1,2)

s = sum(map(lambda x:1/x,fenmu))

print(s)

# 官网参考答案

n =17

ls = sum([1/i for i in range(n,0,-2)])

print(ls)

'''

题目077：循环输出列表

'''

def tm077():

'''

【个人备注】：无需动脑

'''

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

for i in l:

print(i)

'''

题目078：找到年龄最大的人，并输出。

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

'''

def tm078():

'''

【个人备注】：官网的答案也基本一样。

'''

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

name,age='',0

for p in person.keys():

if person.get(p)>age:

name,age=p,person.get(p)

print(name,age)

'''

题目080：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？

'''

def tm080():

'''

【个人备注】：不考虑解方程什么的，问题实际上归结成

“整数n，迭代5次之每次都能得到整数，求n的最小值”问题。

一开始想简单了，以为最后的猴子只拿一个，代进去一算发现不是整数。

于是直接暴力输入，一个个试直到3121，得到整除。

当然也可以反着试，假设最后一个猴子拿到n个桃子，remain = lambda t:t/4\*5+1，

一个个试直到1020，得到整除，也能得出结论，最开始的猴子拿了3121个桃子。

'''

for total in range(10000):

t = total # 沙滩上有t个桃子

remain = lambda t:(t-1)/5\*4 # 每次分桃后剩余桃子。

for i in range(5):

t = remain(t)

if t%1!=0:break # 如果不是整数，说明不符合题意

else:

print(total,t) # 5次都能得到整数，第一个猴子3121，五个猴子拿完沙滩剩余1020个。

break

'''

题目081：809\*??=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。

'''

def tm081():

'''

假设??为x，因为8\*x<100，所以x<13。

实际上因为9\*x>100，得到x>11，很明显x就是12，题意给的过于充分了。

程序如下：

'''

l = lambda x:len(str(x))

for i in range(20):

if l(809\*i)==4 and l(8\*i)==2 and l(9\*i)==3:

x = i

print(x)

print(809\*x==800\*x+9\*x)

print(809\*x)

'''

题目082：八进制转换为十进制

'''

def tm082():

'''

【个人备注】：知道了这些，你就可以随意转了。

'''

print(bin(10)) #十转二

print(oct(10)) #十转八

print(hex(10)) #十转16

print(int('10',8)) #八转十

print(int('10',2)) #二转十

print(int('10',16)) #16转十

'''

题目083：求0—7所能组成的奇数个数。

'''

def tm083():

'''

【个人备注】：没说组成几位数或是否重复使用。假设1-8位都可以，且不能重复使用。

直接用排列函数，累加然后去重，就得到答案了。

'''

s = [i for i in '01234567']

import itertools

arr = []

for i in range(1,9):

a = list(itertools.permutations(s,i)) # 长度1-8左右排列

l = list(map(lambda x:int(''.join(x)),a)) # 整理成数字形式（避免出现02这种情况，02实际上就是2）

arr+=l

print(i,len(l))

arr1 = set(arr) # 去重

arr2 = list(filter(lambda x:x%2==1,arr1)) # 只留奇数

print(len(arr),len(arr1),len(arr2)) # 答案是46972

def tm083\_1():

'''

【个人备注】： 因为也不知道对错，又用穷举法验证验证了一下

'''

count = 0

for i in range(76543211): # 能组成的最大数字也就是76543210了

s = str(i) # 转换成文本形式s

if '8' in s or '9' in s: # s中不包含8和9

continue

else:

cs = set([c for c in s])# s中的数字去重，如果去重后和去重前长度一致，说明数字没有重复使用

if len(s)==len(cs) and s[-1] in '1357': # 各位不重复且是奇数

count+=1

if i%100000==0:print(i,count) # 每10万个输出一下结果，避免程序卡死发现不了。

print(count) # 公司电脑比较差劲，跑了2分钟多，也出结果了46972。

'''

题目084：连接字符串。

'''

def tm084():

'''

【个人备注】： join的用法，之前都已经用过很多次了

'''

pass

'''

题目085：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。

'''

def tm085():

'''

【个人备注】：挨个试直到整除为止即可。

'''

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

for i in range(1,61):

if int('9'\*i)%x==0:

print(i)

break

else:

print('no way')

'''

题目086：两个字符串连接程序。

'''

def tm086():

# 直接‘+’就行

pass

'''

题目087：回答结果（结构体变量传递）。

'''

def tm087():

'''

没题目没得做

'''

pass

'''

题目088：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＊。

'''

def tm088():

# 没啥说的

for i in [1,4,5,14,22]:

print('\*'\*i)

'''

题目089：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。

'''

def tm089():

x,c = 1234,5

q,b,s,g = x//1000,x//100%10,x//10%10,x%10

s = (g+c)%10\*1000+(s+c)%10\*100+(b+c)%10\*10+(q+c)%10

print(s)

'''

题目090：列表使用实例。

'''

def tm090():

# 没有具体要求

pass

'''

题目091：时间函数举例1。

题目092：时间函数举例2。

题目093：时间函数举例3。

题目095：字符串日期转换为易读的日期格式。

'''

def tm091(): #tm092、tm093、tm095

# 参看tm016

pass

'''

题目094：时间函数举例4,一个猜数游戏，判断一个人反应快慢。

'''

def tm094():

import time,random

print('《猜大小0-1000之间》')

x = random.randint(0,1000)

flag = input('是否开始(y/n)：')

if flag=='y':

s = time.time()

while 1:

m = int(input('请输入数字：'))

if m>x:

print('大了')

elif m<x:

print('小了')

else:

print('bingo!')

break

e = time.time()

print('耗时%.2f秒'%(e-s))

print(time.sleep(5))

'''

题目096：计算字符串中子串出现的次数。

'''

def tm096():

'''

【个人备注】：用count就行了

'''

x = 'ababaabbaaa'

print(x.count('ab'))

'''

题目097：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。

'''

def tm097():

'''

【个人备注】：保存文件的方法，记住即可。

with .. as ..打开会自动关闭。

其他方式打开，别忘了通过代码关闭。

'''

path = 'd:/test.txt'

with open(path,'w+') as f:f.write('')

while 1:

c = input()

if c=='#':

break

else:

with open(path,'a+') as f:f.write(c)

'''

题目098：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。

'''

def tm098():

'''

【个人备注】：字符串大写

'''

c = input()

c = c.upper()

with open('d:/test.txt','w+') as f:f.write(c)

'''

题目099：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。

'''

def tm099():

'''

【个人备注】：读写文件

'''

with open('d:/a.txt','r+') as f:a=f.read()

with open('d:/b.txt','r+') as f:b=f.read()

with open('d:/c.txt','w+') as f:f.write(a+b)

'''

题目100：列表转换为字典。

'''

def tm100():

'''

【个人备注】：终于最后一题完事~，没啥说的。

'''

l = ['ak17','b51','b52','#64']

d = {}

for i in range(len(l)):

d[str(i)]=l[i]

print(d)

# {'1': 'b51', '3': '#64', '2': 'b52', '0': 'ak17'}