**原网页：https://www.runoob.com/python/python-100-examples.html**

**1-2基础——3-6常见数组表——7-11时间有关函数——12-19遍历求解——20-27字符串变化——28-34数学应用——35-36字典——37-37函数——**

**40-46画图——47-49其他**

**实例001：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)

============

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002：ord/chr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))

a,b,c=97,98,99 #ord(a),ord(b),ord(c)

JIA=[a,b,c]

flag = 0

for x in range(97,100):

for z in range(97,100):

if x != a and x != c and z != c and x != z:

flag = 1

break

if flag:

break

y=sum(JIA)-x-z

YI=[x,y,z]

print('x:%s,y:%s,z:%s'%(chr(x),chr(y),chr(z)))

**实例003：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004：九九乘法表**

\*\*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005：斐波那契数列I**

\*\*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,b=2,1

c=a/b

for i in range(19):

a,b=a+b,a

c+=a/b

print(a,b,c)

**实例006：斐波那契数列II**

\*\*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

res,a,b=0,1,1

for i in range(num-1):

a,b=b,a+b

print(a)

#前n个数

def fib(n):

if n == 1:

return [1]

if n == 2:

return [1, 1]

fibs = [1, 1]

for i in range(2, n):

fibs.append(fibs[-1] + fibs[-2])

return fibs

print(fib(int(input())))

#引申养兔子第2月末可繁殖

month=int(input('繁殖几个月？： '))

month\_0,month\_1,month\_elder=1,0,0

for i in range(month):

month\_0,month\_1,month\_elder=month\_elder+month\_1,month\_0,month\_elder+month\_1

print('第%d月末共'%(i+1),month\_0+month\_1+month\_elder,'对兔子')

print('新生兔：',month\_0,end=' ')

print('1月兔：',month\_1,end=' ')

print('2月兔：',month\_elder)

**实例007：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，或用datetime库

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)

import datetime

f = input('请输入年月日（格式：YYYY-MM-DD）:')

yy,mm,dd = f.split('-')

d=datetime.date(int(yy),int(mm),int(dd))

d0=datetime.date(int(yy),1,1)

delta = d-d0+datetime.timedelta(1)

print(str(delta).split(',')[0])

**实例008：dateutil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009：time—猜数字计时**

\*\*题目：\*\*时间函数举例4，猜数字。

\*\*程序分析：\*\*如何浪费时间。

import time

import random

I=[]

i=0

play\_it = input('你想玩猜数字游戏吗.(Y or N)：')

while play\_it in ['Y','y']:

c = input('输入你的角色:')

I.append([c])

x = random.randint(1,101)

start = time.perf\_counter()

guess = int(input('请输入你猜的数字:\n'))

I[i].append(guess)

while guess != x:

if guess > x:

print('再小一点！')

guess = int(input('请输入你猜的数字:\n'))

I[i].append(guess)

elif guess < x:

print('再大一点！')

guess = int(input('请输入你猜的数字:\n'))

I[i].append(guess)

else:

guess = int(input('你输入的数字是?'))

I[i].append(guess)

end = time.perf\_counter()

print ('你花了 %.2f 秒猜到了数字 %d.' % (end-start,x))

if end-start < 15:

print ('真聪明!')

elif end-start < 25:

print ('还可以再快一点点!')

else:

print ('继续努力吧!')

print ('%s ,你猜过的数字有 %s, 猜每个数字用了 %.2f 秒'% \

(c,I[i][1:],(end-start)/(len(I[i])-1)))

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

i += 1

**实例010：time—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：datetime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012：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)

##

I = []

for i in range(101):

for j in range(2,round(i\*\*0.5)+1):

if i%j==0:

break

else:

I.append(i)

print(I,len(I))

##

def func\_get\_prime(n):

return filter(lambda x: not [x%i for i in range(2, int(x\*\*0.5)+1) if x%i ==0], range(2,n+1))

print (list(func\_get\_prime(100)),len(list(func\_get\_prime(100))))

**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=233\*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：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016：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017：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。#m2-n2=168,求n2

#m>=n+1,m2>=(n+1)2,168+n2>=(n+1)2,168>=2n+1,n<=83.5

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：

for i in range(84):

if (i\*\*2+168)\*\*0.5==int((i\*\*2+168)\*\*0.5):

print(i\*\*2-100)

**实例018：条件求解方程**

\*\*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。

for i in range(10,100):

b = i \* 809

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

print(i,b)

##

for i in range(10,100):

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

continue

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

print(i,809\*i)

break

##

x=10

while x<100:

x+=1

if len(str(809\*x))==4 and len(str(8\*x))==2 and len(str(9\*x))==3:

print(x,809\*x)

**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={1}

for i in range(2,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)):

print(i,factor(i))

**实例020：统计字符串构成**

\*\*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,num,spa,oth=0,0,0,0

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,'','digit:',num,'','alpha:',alp,'','other:',oth)

**实例021：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)

I=[int(a)]

for i in range(4):

a+=a[0]

I.append(int(a))

print(sum(I))

**实例022：n个9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))

x=int(input('输入一个数字:'))

n=0

a=9

while n<10000:

n+=1

if a%x == 0:

print(a,len(str(a),'个9可以整除','x'))

break

else:

a=a\*10+9

**实例023：菱形**

\*\*题目：\*\*打印出如下图案（菱形）:

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

\*\*程序分析：\*\*递归调用即可。

def draw(num):

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

a=i\*'\*'

print(a.center(2\*num+1,' '))

for i in range(2\*num-3,0,-2):

a=i\*'\*'

print(a.center(2\*num+1,' '))

draw(10)

**实例024：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025：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026：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]

li[li.index(max(li))],li[0]=li[0],li[li.index(max(li))]

print(li)

**实例027：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)

cut=int(input('rotate:'))

deq.rotate(cut)

print(list(deq))

a,b = li[:(len(li)-cut)],li[(len(li)-cut):]

print(b+a)

**实例028：奇偶倒数求和**

\*\*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

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

if n % 2 == 0:

sum = dcall(peven,n)

else:

sum = dcall(podd,n)

print (sum)

##

def hs(num):

s=0

if num%2==0:

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

s += 1/i

else:

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

s += 1/i

return s

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

print(hs(n))

**实例029：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

def

for i in range(4,1,-1):

res=i\*res+1

print(i,res)

print(res)

##

def jc(n):

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

res=0

for i in range(4):

res += jc(i+1)

print(res)

##

n,s,t = 0,0,1

for n in range(1,21):

t \*= n

s += t

print (s)

##

s = 0

l = range(1,21)

def op(x):

r = 1

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

r \*= i

return r

print(sum(map(op,l)))

**实例030：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031：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：第5只猴分的5份其中每份1个，且多一个，共6个

共有5次判断，每次都是判断剩下的桃子数量减一后是否可以除尽5？

办法是引入count作为计数器，满足以上条件一次则count-1，直至count=0；

如果中间有无法除尽的情况(num - 1) % 5 != 0，则程序跳出判断，返回-1值

需要思考每次猴子连吃带拿，剩下的桃子数量变动是否符合递归？

每一次分配桃子都有共同的地方：减去1，再除以5，可以除尽

符合递归的特征：

描述第count次分桃子和第count-1次分桃子，桃子数量num之间存在以下联系

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

只有当if判断num能整除5，则输出下一个猴子分桃时看到的桃子数量

return consume(count - 1, num)。

i,j,x = 0,1,0

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)

##

def consume(count, num):

if count == 0:

return 1

elif (num - 1) % 5 == 0:

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

return consume(count - 1, num)

count = int(input('请输入猴子个数：'))

num = 1

while count:

num+=1

if consume(count,num) == 1:

print('原来最少有 %d 个桃子'%num)

break

**实例032：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

n=int(input('请输入要计算的次数:'))

height=100

tour = []

def tour\_sum(num):

long = height

for i in range(num):

tour.append(long)

long /= 2

tour.append(long)

print('第%d次回弹高度是%f米' % (num,long))

return sum(tour)-long

print('第%d次落地时共走路程%f米'% (n,tour\_sum(n)))

**实例033：“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)

I = int(input('净利润:'))

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

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

bonus = 0

for i in range(6):

if I > arr[i]:

r += (I-arr[i])\*rate[i]

print((I-arr[i])\*rate[i])

I = arr[i]

print(r)

**实例034：围圈报数**

\*\*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。

nmax = 50

n = int(input('请输入总人数:'))

num = []

for i in range(n):

num.append(i + 1)

i,k,m = 0,0,0

while m < n - 1: #m为删除的人数，n为总人数，k是步数，i是座位号

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

if k == 3:

num[i] = 0

k = 0

m += 1

i += 1

if i == n : i = 0

print('座位号是 %s 的人留到了最后'%sum(num))

m,n=input('请输入人数和喊点数：').split(',')

m,n=int(m),int(n)

li=list(range(1,m+1))

count = 0

for i in range(1,(m-1)\*n+1):

count+=1

if i%n==0:

del (li[count-1])

## print(li)

count-=1

if count==len(li):

count=0

print('座位号是 %s 的人留到了最后'%li[0])

**实例035：字典字母识词**

\*\*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036：字典排序最值**

\*\*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。

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037：函数输入和输出**

\*\*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])

input\_stu(student)

print (student)

output\_stu(student)

**实例038：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039：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

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040：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041：画图**

\*\*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042：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043：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044：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045：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046：画组合图形**

\*\*题目：\*\*一个最优美的图案（黄底）。

\*\*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047：访问类成员**

\*\*题目：\*\*回答结果（结构体变量传递）。

\*\*程序分析：\*\*无。

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048：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049：磁盘读写+合并排序字母**

\*\*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。

def create\_txt(filename):

fp = open(filename,"w+")

ch = input('输入字符串:\n').upper()

while ch != '#':

fp.write(ch)

## fp.write('\n')

fp.seek(0,0)

print(fp.read())

ch = input('').upper()

fp.close()

def merge\_txt():

with open('111.txt') as fp:

a = fp.read()

with open('222.txt') as fp:

b = fp.read()

with open('333.txt','w+') as fp:

l = list(a + b)

l.sort()

s = ''

s = s.join(l)

fp.write(s)

fp.seek(0,0)

print(fp.read())

create\_txt('111.txt')

create\_txt('222.txt')

merge\_txt()