

Python 练习实例 1

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：可填在百位、十位、个位的数字都是1、2、3、4。组成所有的排列后再去 掉不满足条件的排列。

程序源代码：

实例(Python 2.0+)

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
for i in range(1,5):
    for j in range(1,5):
        for k in range(1,5):
            if( i != k ) and ( i != j ) and ( j != k ):
                print i,j,k
```

以上实例输出结果为：

```
1 2 3
1 2 4
1 3 2
1 3 4
1 4 2
1 4 3
2 1 3
2 1 4
2 3 1
2 3 4
2 4 1
2 4 3
3 1 2
3 1 4
3 2 1
3 2 4
3 4 1
3 4 2
4 1 2
4 1 3
4 2 1
4 2 3
4 3 1
```

Python 练习实例 2



Python 100 例

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：请利用数轴来分界，定位。注意定义时需把奖金定义成长整型。

程序源代码：

实例(Python 2.0+)

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
i = int(raw_input('净利润:'))
arr = [1000000, 600000, 400000, 200000, 100000, 0]
rat = [0.01, 0.015, 0.03, 0.05, 0.075, 0.1]
r = 0
for idx in range(0, 6):
    if i > arr[idx]:
        r += (i - arr[idx]) * rat[idx]
        print (i - arr[idx]) * rat[idx]
        i = arr[idx]
print r
```

以上实例输出结果为：

```
净利润:120000
1500.0
10000.0
11500.0
```

Python 练习实例 3



Python 100 例

题目：一个整数，它加上 100 和加上 268 后都是一个完全平方数，请问该数是多少？

程序分析：在 10000 以内判断，将该数加上 100 后再开方，加上 268 后再开方，如果开方后的结果满足如下条件，即是结果。请看具体分析：

程序源代码：

实例(Python 2.0+)

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import math
for i in range(10000):
    #转化为整型值
    x = int(math.sqrt(i + 100))
    y = int(math.sqrt(i + 268))
    if(x * x == i + 100) and (y * y == i + 268):
        print i
```

以上实例输出结果为：

```
21
261
1581
```

Python 练习实例 4



Python 100 例

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：以 3 月 5 日为例，应该先把前两个月的加起来，然后再加上 5 天即本年的第几天，

特殊情况，闰年且输入月份大于 2 时需考虑多加一天：

程序源代码：

实例(Python 2.0+)

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
year = int(raw_input('year:\n'))
month = int(raw_input('month:\n'))
day = int(raw_input('day:\n'))
months = (0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334)
if 0 < month <= 12:
    sum = months[month - 1]
```

```

else:
    print 'data error'
    sum += day leap = 0
    if (year % 400 == 0) or ((year % 4 == 0) and (year % 100 != 0)):
        leap = 1 if (leap == 1) and (month > 2):
            sum += 1
    print 'it is the %dth day.' % sum

```

以上实例输出结果为：

```

year:
2015
month:
6
day:
7
it is the 158th day.

```

Python 练习实例 5



[Python 100 例](#)

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：我们想办法把最小的数放到 x 上，先将 x 与 y 进行比较，如果 x>y 则将 x 与 y 的值进行交换，然后再用 x 与 z 进行比较，如果 x>z 则将 x 与 z 的值进行交换，这样能使 x 最小。

程序源代码：

```

#!/usr/bin/python
# -*- coding: UTF-8 -*-

l = []
for i in range(3):
    x = int(raw_input('integer:\n'))
    l.append(x)
l.sort()
print l

```

以上实例输出结果为：

```

integer:
8
integer:

```

```
5
integer:
6
[5, 6, 8]
```

Python 练习实例 6



题目：斐波那契数列。

程序分析：斐波那契数列（Fibonacci sequence），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、.....。

在数学上，费波那契数列是以递归的方法来定义：

```
F0 = 0      (n=0)
F1 = 1      (n=1)
Fn = F[n-1] + F[n-2] (n>2)
```

程序源代码：

方法一

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

def fib(n):
    a,b = 1,1
    for i in range(n-1):
        a,b = b,a+b
    return a

# 输出了第 10 个斐波那契数列
print fib(10)
```

方法二

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

# 使用递归
```

```
def fib(n):  
    if n==1 or n==2:  
        return 1  
    return fib(n-1)+fib(n-2)  
  
# 输出了第 10 个斐波那契数列  
print fib(10)
```

以上实例输出了第 10 个斐波那契数列，结果为：

55

方法三

如果你需要输出指定个数的斐波那契数列，可以使用以下代码：

```
#!/usr/bin/python  
# -*- coding: UTF-8 -*-  
  
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  
  
# 输出前 10 个斐波那契数列  
print fib(10)
```

以上程序运行输出结果为：

[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

Python 练习实例 7



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[:]。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = [1, 2, 3]
b = a[:]
print b
```

以上实例输出结果为：

```
[1, 2, 3]
```

Python 练习实例 8



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。

程序源代码：

9*9 乘法口诀

```
#!/usr/bin/python # -*- coding: UTF-8 -*- for i in range(1, 10): print
for j in range(1, i+1): print "%d*%d=%d" % (i, j, i*j),
```

以上实例输出结果为：

```
1*1=1
2*1=2 2*2=4
3*1=3 3*2=6 3*3=9
4*1=4 4*2=8 4*3=12 4*4=16
5*1=5 5*2=10 5*3=15 5*4=20 5*5=25
6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36
7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49
8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64
9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

Python 练习实例 9



题目：暂停一秒输出。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import time

myD = {1: 'a', 2: 'b'}
for key, value in dict.items(myD):
    print key, value
    time.sleep(1) # 暂停 1 秒
```

以上实例输出结果为(会有停顿效果)：

```
1 a
2 b
```

Python 练习实例 10



题目：暂停一秒输出，并格式化当前时间。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import time

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

# 暂停一秒
time.sleep(1)

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

以上实例输出结果为：

```
2015-10-21 17:48:40
2015-10-21 17:48:41
```

Python 练习实例 11



题目：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？

程序分析：兔子的规律为数列 1,1,2,3,5,8,13,21....

程序源代码：

实例(Python 2.0+)

```
#!/usr/bin/python # -*- coding: UTF-8 -*- f1 = 1 f2 = 1 for i in range(1,22): print '%12ld %12ld' % (f1,f2), if (i % 3) == 0: print ' ' f1 = f1 + f2 f2 = f1 + f2
```

以上实例输出结果为：

1	1	2	3	5	8
13	21	34	55	89	144
233	377	610	987	1597	2584
4181	6765	10946	17711	28657	46368
75025	121393	196418	317811	514229	832040
1346269	2178309	3524578	5702887	9227465	14930352
24157817	39088169	63245986	102334155	165580141	267914296

Python 练习实例 12



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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

h = 0
leap = 1
from math import sqrt
```

```
from sys import stdout
for m in range(101,201):
    k = int(sqrt(m + 1))
    for i in range(2,k + 1):
        if m % i == 0:
            leap = 0
            break
    if leap == 1:
        print '%-4d' % m
        h += 1
        if h % 10 == 0:
            print ''
    leap = 1
print 'The total is %d' % h
```

以上实例输出结果为：

```
101
103
107
109
113
127
131
137
139
149

151
157
163
167
173
179
181
191
193
197

199
The total is 21
```

Python 练习实例 13

Python 100 例

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 $153=1^3+5^3+3^3$ 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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

for n in range(100,1000):
    i = n / 100
    j = n / 10 % 10
    k = n % 10
    if n == i ** 3 + j ** 3 + k ** 3:
        print n
```

以上实例输出结果为：

```
153
370
371
407
```

Python 练习实例 14

Python 100 例

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

程序分析：对 n 进行分解质因数，应先找到一个最小的质数 k，然后按下述步骤完成：

- (1)如果这个质数恰等于 n，则说明分解质因数的过程已经结束，打印出即可。
- (2)如果 $n < k$ ，但 n 能被 k 整除，则应打印出 k 的值，并用 n 除以 k 的商,作为新的正整数你 n, 重复执行第一步。
- (3)如果 n 不能被 k 整除，则用 k+1 作为 k 的值,重复执行第一步。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
```

```
def reduceNum(n):
    print '{} = '.format(n),
    if not isinstance(n, int) or n <= 0 :
        print '请输入一个正确的数字 !'
        exit(0)
    elif n in [1] :
        print '{}'.format(n)
    while n not in [1] : # 循环保证递归
        for index in xrange(2, n + 1) :
            if n % index == 0:
                n /= index # n 等于 n/index
                if n == 1:
                    print index
                else : # index 一定是素数
                    print '{} *'.format(index),
                break
reduceNum(90)
reduceNum(100)
```

以上实例输出结果为:

```
90 = 2 * 3 * 3 * 5
100 = 2 * 2 * 5 * 5
```

Python 练习实例 15



[Python 100 例](#)

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：程序分析：(a>b)?a:b 这是条件运算符的基本例子。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

score = int(raw_input('input score:\n'))
if score >= 90:
    grade = 'A'
elif score >= 60:
    grade = 'B'
```

```
else:
    grade = 'C'

print '%d belongs to %s' % (score, grade)
```

以上实例输出结果为：

```
input score:
100
100 belongs to A
```

Python 练习实例 16



Python 100 例

题目：输出指定格式的日期。

程序分析：使用 `datetime` 模块。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import datetime

if __name__ == '__main__':

    # 输出今日日期，格式为 dd/mm/yyyy。更多选项可以查看 strftime() 方法
    print(datetime.date.today().strftime('%d/%m/%Y'))

    # 创建日期对象
    miyazakiBirthDate = datetime.date(1941, 1, 5)

    print(miyazakiBirthDate.strftime('%d/%m/%Y'))

    # 日期算术运算
    miyazakiBirthNextDay = miyazakiBirthDate + datetime.timedelta(days=1)

    print(miyazakiBirthNextDay.strftime('%d/%m/%Y'))

    # 日期替换
    miyazakiFirstBirthday = miyazakiBirthDate.replace(year=miyazakiBirthDate.
year + 1)
```

```
print(miyazakiFirstBirthday.strftime('%d/%m/%Y'))
```

以上实例输出结果为:

```
21/10/2015
05/01/1941
06/01/1941
05/01/1942
```

Python 练习实例 17



Python 100 例

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 语句,条件为输入的字符不为'\n'。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import string
s = raw_input('input a string:\n')
letters = 0
space = 0
digit = 0
others = 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)
```

以上实例输出结果为:

```
input a string:
```

```
runoob
char = 6,space = 0,digit = 0,others = 0
```

Python 练习实例 18



题目：求 $s=a+aa+aaa+aaaa+aa\dots a$ 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)，几个数相加有键盘控制。

程序分析：关键是计算出每一项的值。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

Tn = 0
Sn = []
n = int(raw_input('n = :\n'))
a = int(raw_input('a = :\n'))
for count in range(n):
    Tn = Tn + a
    a = a * 10
    Sn.append(Tn)
    print Tn

Sn = reduce(lambda x,y : x + y,Sn)
print Sn
```

以上实例输出结果为：

```
n = :
2
a = :
4
4
44
48
```

Python 练习实例 19

Python 100 例

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：请参照程序 [Python 练习实例 14](#)。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from sys import stdout
for j in range(2,1001):
    k = []
    n = -1
    s = j
    for i in range(1,j):
        if j % i == 0:
            n += 1
            s -= i
            k.append(i)

    if s == 0:
        print j
        for i in range(n):
            stdout.write(str(k[i]))
            stdout.write(' ')
        print k[n]
```

以上实例输出结果为：

```
6
1 2 3
28
1 2 4 7 14
496
1 2 4 8 16 31 62 124 248
```

Python 练习实例 20

Python 100 例

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

程序源代码：

Python 实例

```
#!/usr/bin/python # -*- coding: UTF-8 -*- tour = [] height = [] hei = 100.0 # 起始高度 tim = 10 # 次数 for i in range(1, tim + 1): tour.append(hei) hei /= 2 height.append(hei) print('总高度: tour = {0}'.format(sum(tour))) print('第 10 次反弹高度: height = {0}'.format(height[-1]))
```

以上实例输出结果为：

```
总高度: tour = 199.8046875
第 10 次反弹高度: height = 0.09765625
```

Python 练习实例 21



Python 100 例

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：采取逆向思维的方法，从后往前推断。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

x2 = 1
for day in range(9, 0, -1):
    x1 = (x2 + 1) * 2
    x2 = x1
print x1
```

以上实例输出结果为：

```
1534
```

Python 练习实例 22



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。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

for i in range(ord('x'),ord('z') + 1):

    for j in range(ord('x'),ord('z') + 1):

        if i != j:

            for k in range(ord('x'),ord('z') + 1):

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

                    if (i != ord('x')) and (k != ord('x')) and (k != ord('z')):

                        print 'order is a -- %s\t b -- %s\t c--%s' % (chr(i),chr

(j),chr(k))
```

以上实例输出结果为：

```
order is a -- z    b -- x    c--y
```

Python 练习实例 23



题目：打印出如下图案（菱形）：

```
    *
   ***
  *****
 *****
 *****
   ***
    *
```

程序分析：先把图形分成两部分来看待，前四行一个规律，后三行一个规律，利用双重 for 循环，第一层控制行，第二层控制列。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from sys import stdout
for i in range(4):
    for j in range(2 - i + 1):
        stdout.write(' ')
    for k in range(2 * i + 1):
        stdout.write('*')
    print

for i in range(3):
    for j in range(i + 1):
        stdout.write(' ')
    for k in range(4 - 2 * i + 1):
        stdout.write('*')
    print
```

以上实例输出结果为：

```
    *
   ***
  *****
 *****
 *****
   ***
    *
```

Python 练习实例 24

Python 100 例

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：请抓住分子与分母的变化规律。

程序源代码：

方法一：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = 2.0
b = 1.0
s = 0
for n in range(1,21):
    s += a / b
    t = a
    a = a + b
    b = t
print s
```

方法二：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = 2.0
b = 1.0
s = 0.0
for n in range(1,21):
    s += a / b
    b,a = a , a + b
print s

s = 0.0
for n in range(1,21):
    s += a / b
    b,a = a , a + b
print s
```

方法三：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = 2.0
b = 1.0
l = []
for n in range(1,21):
    b,a = a,a + b
    l.append(a / b)
print reduce(lambda x,y: x + y,l)
```

以上实例输出结果为：

```
32.6602607986
```

Python 练习实例 25



题目：求 $1+2!+3!+\dots+20!$ 的和。

程序分析：此程序只是把累加变成了累乘。

程序源代码：

方法一：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

n = 0
s = 0
t = 1
for n in range(1,21):
    t *= n
    s += t
print '1! + 2! + 3! + ... + 20! = %d' % s
```

方法二:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

s = 0
l = range(1,21)
def op(x):
    r = 1
    for i in range(1,x + 1):
        r *= i
    return r
s = sum(map(op,l))
print '1! + 2! + 3! + ... + 20! = %d' % s
```

以上实例输出结果为:

```
1! + 2! + 3! + ... + 20! = 2561327494111820313
```

Python 练习实例 26



题目: 利用递归方法求 5!。

程序分析: 递归公式: $fn=fn_1*4!$

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

def fact(j):
    sum = 0
    if j == 0:
        sum = 1
    else:
        sum = j * fact(j - 1)
    return sum

for i in range(5):
    print '%d! = %d' % (i,fact(i))
```

以上实例输出结果为：

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
```

Python 练习实例 27



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：无。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

def output(s,l):

    if l==0:

        return

    print (s[l-1])

    output(s,l-1)

s = raw_input('Input a string:')

l = len(s)

output(s,l)
```

以上实例输出结果为：

```
Input a string:abcde
```

e
d
c
b
a

Python 练习实例 28



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， he 说是 10 岁。请问第五个人多大？

程序分析：利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人岁数，需知道第四人的岁数，依次类推，推到第一人（10 岁），再往回推。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

def age(n):
    if n == 1: c = 10
    else: c = age(n - 1) + 2
    return c
print age(5)
```

以上实例输出结果为：

18

Python 练习实例 29



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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

x = int(raw_input("请输入一个数:\n"))
a = x / 10000
b = x % 10000 / 1000
c = x % 1000 / 100
d = x % 100 / 10
e = x % 10

if a != 0:
    print "5 位数: ",e,d,c,b,a
elif b != 0:
    print "4 位数: ",e,d,c,b,
elif c != 0:
    print "3 位数: ",e,d,c
elif d != 0:
    print "2 位数: ",e,d
else:
    print "1 位数: ",e
```

以上实例输出结果为：

```
请输入一个数:
23459
5 位数:  9 5 4 3 2
请输入一个数:
3472
4 位数:  2 7 4 3
```

Python 练习实例 30



Python 100 例

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：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = int(raw_input("请输入一个数字:\n"))
x = str(a)
flag = True

for i in range(len(x)/2):
    if x[i] != x[-i - 1]:
        flag = False
        break
if flag:
    print "%d 是一个回文数!" % a
else:
    print "%d 不是一个回文数!" % a
```

以上实例输出结果为:

```
请输入一个数字:
12321
12321 是一个回文数!
```

Python 练习实例 31



[Python 100 例](#)

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: 用情况语句比较好, 如果第一个字母一样, 则判断用情况语句或 if 语句判断第二个字母。。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

letter = raw_input("please input:")
#while letter != 'Y':
if letter == 'S':
    print ('please input second letter:')
    letter = raw_input("please input:")
    if letter == 'a':
```

```

        print ('Saturday')
    elif letter == 'u':
        print ('Sunday')
    else:
        print ('data error')

elif letter == 'F':
    print ('Friday')

elif letter == 'M':
    print ('Monday')

elif letter == 'T':
    print ('please input second letter')
    letter = raw_input("please input:")

    if letter == 'u':
        print ('Tuesday')
    elif letter == 'h':
        print ('Thursday')
    else:
        print ('data error')

elif letter == 'W':
    print ('Wednesday')
else:
    print ('data error')

```

以上实例输出结果为：

```

please input:S
please input second letter:
please input:a
Saturday

```

Python 练习实例 32

 [Python 100 例](#)

题目：按相反的顺序输出列表的值。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = ['one', 'two', 'three']
for i in a[::-1]:
    print i
```

以上实例输出结果为：

```
three
two
one
```

Python 练习实例 33

 [Python 100 例](#)

题目：按逗号分隔列表。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

L = [1,2,3,4,5]
s1 = ','.join(str(n) for n in L)
print s1
```

以上实例输出结果为：

```
1,2,3,4,5
```

Python 练习实例 34

 [Python 100 例](#)

题目：练习函数调用。

程序分析：无。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

def hello_world():
    print 'hello world'

def three_hellos():
    for i in range(3):
        hello_world()
if __name__ == '__main__':
    three_hellos()
```

以上实例输出结果为:

```
hello world
hello world
hello world
```

Python 练习实例 35



题目: 文本颜色设置。

程序分析: 无。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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
```

以上实例输出结果为：

警告的颜色字体? # 浅黄色

Python 练习实例 36



题目：求 100 之内的素数。

程序分析：无。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

# 输出指定范围内的素数

# take input from the user

lower = int(input("输入区间最小值："))

upper = int(input("输入区间最大值："))

for num in range(lower,upper + 1):

    # 素数大于 1

    if num > 1:

        for i in range(2,num):

            if (num % i) == 0:

                break

        else:
```

```
print(num)
```

以上实例输出结果为：

输入区间最小值：1

输入区间最大值：100

2

3

5

7

11

13

17

19

23

29

31

37

41

43

47

53

59

61

67

71

73

79

83

89

97

Python 练习实例 37



题目：对 10 个数进行排序。

程序分析：可以利用选择法，即从后 9 个比较过程中，选择一个最小的与第一个元素交换，下次类推，即用第二个元素与后 8 个进行比较，并进行交换。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

if __name__ == "__main__":

    N = 10

    # input data

    print 'please input ten num:\n'

    l = []

    for i in range(N):

        l.append(int(raw_input('input a number:\n')))

    print

    for i in range(N):

        print l[i]

    print
```

```
# sort ten num

for i in range(N - 1):

    min = i

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

        if l[min] > l[j]:min = j

    l[i],l[min] = l[min],l[i]

print 'after sorted'

for i in range(N):

    print l[i]
```

以上实例输出结果为:

```
please input ten num:
```

```
input a number:
```

```
5
```

```
input a number:
```

```
3
```

```
input a number:
```

```
23
```

```
input a number:
```

```
67
```

```
input a number:
```

```
2
```

```
input a number:
```

```
56
```

input a number:

45

input a number:

98

input a number:

239

input a number:

9

5

3

23

67

2

56

45

98

239

9

after sorted

2

3

5

9

23

45

56

67

98

239

Python 练习实例 38



题目：求一个 3*3 矩阵对角线元素之和。

程序分析：利用双重 for 循环控制输入二维数组，再将 a[i][i]累加后输出。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

if __name__ == '__main__':

    a = []

    sum = 0.0

    for i in range(3):

        a.append([])

        for j in range(3):

            a[i].append(float(raw_input("input num:\n")))

    for i in range(3):

        sum += a[i][i]

    print sum
```

以上实例输出结果为：

```
input num:
78
input num:
34
input num:
23
input num:
34
input num:
56
input num:
33
input num:
12
input num:
21
input num:
2
136.0
```

Python 练习实例 39



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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    # 方法一
    a = [1,4,6,9,13,16,19,28,40,100,0]
    print 'original list is:'
    for i in range(len(a)):
        print a[i]
    number = int(raw_input("insert a new number:\n"))
    end = a[9]
    if number > end:
        a[10] = number
    else:
        for i in range(10):
            if a[i] > number:
                temp1 = a[i]
                a[i] = number
                for j in range(i + 1,11):
                    temp2 = a[j]
                    a[j] = temp1
                    temp1 = temp2
                break
        for i in range(11):
            print a[i]
```

以上实例输出结果为：

```
original list is:
1
4
6
9
13
16
19
28
40
100
0
```

```
insert a new number:  
7  
1  
4  
6  
7  
9  
13  
16  
19  
28  
40  
100
```

Python 练习实例 40



[Python 100 例](#)

题目：将一个数组逆序输出。

程序分析：用第一个与最后一个交换。

程序源代码：

```
#!/usr/bin/python  
# -*- coding: UTF-8 -*-  
  
if __name__ == '__main__':  
    a = [9,6,5,4,1]  
    N = len(a)  
    print a  
    for i in range(len(a) / 2):  
        a[i],a[N - i - 1] = a[N - i - 1],a[i]  
    print a
```

以上实例输出结果为：

```
[9, 6, 5, 4, 1]  
[1, 4, 5, 6, 9]
```



[Python 100 例](#)

← [Python JSON](#)

笔记列表

以下方式也可以将一个数组逆序输出：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = [9,6,5,4,1]
    print a[::-1]
```

Python 练习实例 41



[Python 100 例](#)

题目：模仿静态变量的用法。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

def varfunc():
    var = 0
    print 'var = %d' % var
    var += 1

if __name__ == '__main__':
    for i in range(3):
        varfunc()

# 类的属性
# 作为类的一个属性吧
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()
```

以上实例输出结果为：

```
var = 0
var = 0
var = 0
5
6
7
8
```

Python 练习实例 42



题目：学习使用 `auto` 定义变量的用法。

程序分析：没有 `auto` 关键字，使用变量作用域来举例吧。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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
```

Python 练习实例 43



题目：模仿静态变量(static)另一案例。

程序分析：演示一个 python 作用域使用方法

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

class Num:
    nNum = 1
    def inc(self):
        self.nNum += 1
        print 'nNum = %d' % self.nNum

if __name__ == '__main__':
    nNum = 2
    inst = Num()
    for i in range(3):
        nNum += 1
        print 'The num = %d' % nNum
        inst.inc()
```

以上实例输出结果为：

```
The num = 3
nNum = 2
The num = 4
nNum = 3
The num = 5
nNum = 4
```

Python 练习实例 44 - Python 两个 矩阵相加



两个 3 行 3 列的矩阵，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]]
```

程序分析：创建一个新的 3 行 3 列的矩阵，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。

程序源代码：

源代码：

```
#!/usr/bin/python # -*- coding: UTF-8 -*- X = [[12,7,3], [4 ,5,6], [7 ,
8,9]] Y = [[5,8,1], [6,7,3], [4,5,9]] result = [[0,0,0], [0,0,0], [0,0,
0]] # 迭代输出行 for i in range(len(X)): # 迭代输出列 for j in range(len(X
[0])): result[i][j] = X[i][j] + Y[i][j] for r in result: print(r)
```

执行以上代码，输出结果如下：

```
[17, 15, 4]

[10, 12, 9]

[11, 13, 18]
```

Python 练习实例 45

 [Python 100 例](#)

题目：统计 1 到 100 之和。

程序分析：无

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

tmp = 0
for i in range(1,101):
    tmp += i
print 'The sum is %d' % tmp
```

以上实例输出结果为:

```
The sum is 5050
```

Python 练习实例 46



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

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

TRUE = 1
FALSE = 0
def SQ(x):
    return x * x
print '如果输入的数字小于 50，程序将停止运行。'
again = 1
while again:
    num = int(raw_input('Please input number'))
    print '运算结果为 %d' % (SQ(num))
    if num >= 50:
        again = TRUE
    else:
        again = FALSE
```

以上实例输出结果为:

```
如果输入的数字小于 50，程序将停止运行。
Please input number300
```

运算结果为 90000

Please input number12

运算结果为 144

Python 练习实例 47



题目：两个变量值互换。

程序分析：无

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

def exchange(a,b):
    a,b = b,a
    return (a,b)

if __name__ == '__main__':
    x = 10
    y = 20
    print 'x = %d,y = %d' % (x,y)
    x,y = exchange(x,y)
    print 'x = %d,y = %d' % (x,y)
```

以上实例输出结果为：

```
x = 10,y = 20
x = 20,y = 10
```

Python 练习实例 48



题目：数字比较。

程序分析：无

程序源代码：

```
#!/usr/bin/python
```

```
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    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'
```

以上实例输出结果为：

```
10 小于 20
```

Python 练习实例 49



[Python 100 例](#)

题目：使用 lambda 来创建匿名函数。

程序分析：无

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

MAXIMUM = lambda x,y : (x > y) * x + (x < y) * y
MINIMUM = lambda x,y : (x > y) * y + (x < y) * x

if __name__ == '__main__':
    a = 10
    b = 20
    print 'The largar one is %d' % MAXIMUM(a,b)
    print 'The lower one is %d' % MINIMUM(a,b)
```

以上实例输出结果为：

```
The largar one is 20
The lower one is 10
```

Python 练习实例 50



题目：输出一个随机数。

程序分析：使用 random 模块。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import random

#生成 10 到 20 之间的随机数
print random.uniform(10, 20)
```

以上实例输出结果为：

```
14.4012371357
```

Python 练习实例 51



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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = 077
    b = a & 3
    print 'a & b = %d' % b
    b &= 7
    print 'a & b = %d' % b
```

以上实例输出结果为：

```
a & b = 3
```

```
a & b = 3
```

Python 练习实例 52



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

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = 077
    b = a | 3
    print 'a | b is %d' % b
    b |= 7
    print 'a | b is %d' % b
```

以上实例输出结果为：

```
a | b is 63
```

```
a | b is 63
```

Python 练习实例 53



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

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = 077
    b = a ^ 3
```

```
print 'The a ^ 3 = %d' % b
b ^= 7
print 'The a ^ b = %d' % b
```

以上实例输出结果为:

```
The a ^ 3 = 60
The a ^ b = 59
```

Python 练习实例 54



[Python 100 例](#)

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 $\sim(\sim 0 \ll 4)$
- (3)将上面二者进行 & 运算。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = int(raw_input('input a number:\n'))
    b = a >> 4
    c = ~(~0 << 4)
    d = b & c
    print '%o\t%o' % (a, d)
```

以上实例输出结果为:

```
input a number:
9
11      0
```

Python 练习实例 55



[Python 100 例](#)

题目：学习使用按位取反~。

程序分析：~0=1; ~1=0;

(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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = 234
    b = ~a
    print 'The a\'s 1 complement is %d' % b
    a = ~a
    print 'The a\'s 2 complement is %d' % a
```

以上实例输出结果为：

```
The a's 1 complement is -235
The a's 2 complement is -235
```

Python 练习实例 56



题目：画图，学用 circle 画圆形。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

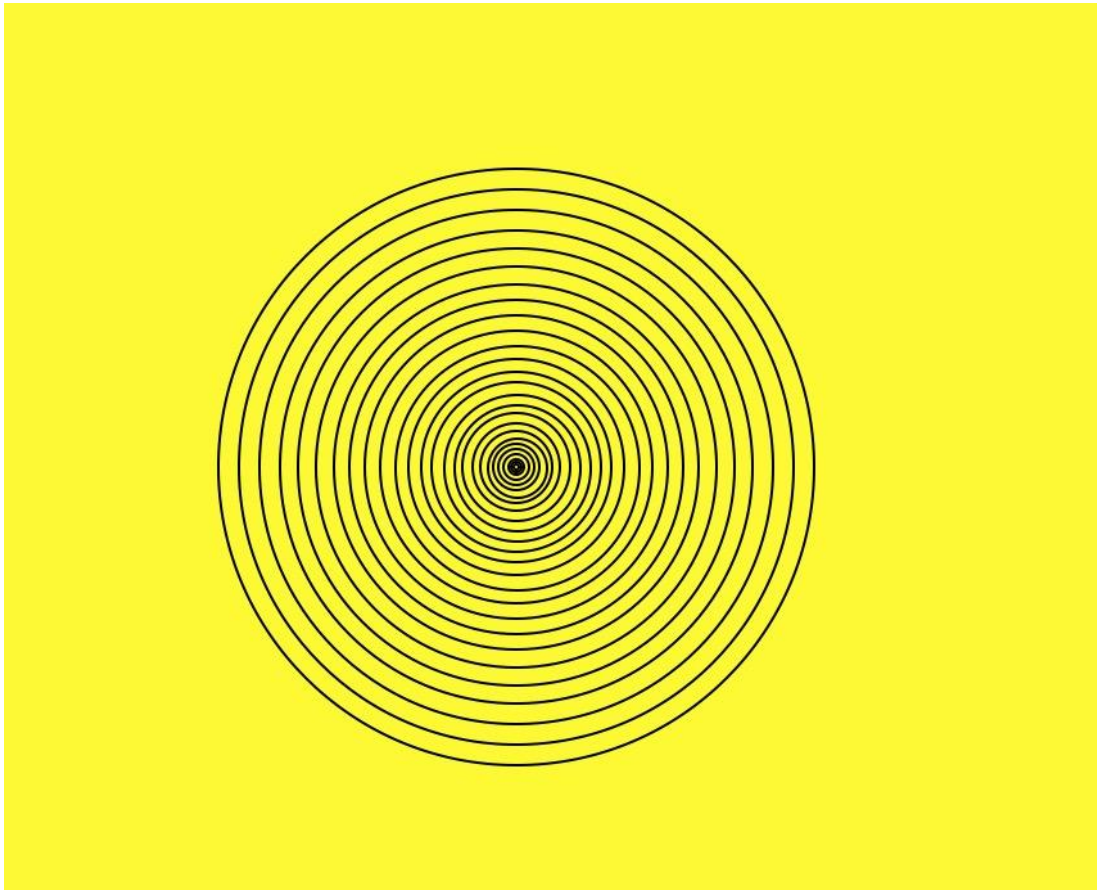
if __name__ == '__main__':
    from Tkinter import *

    canvas = Canvas(width=800, height=600, bg='yellow')
    canvas.pack(expand=YES, fill=BOTH)
    k = 1
    j = 1
```

```
for i in range(0,26):
    canvas.create_oval(310 - k,250 - k,310 + k,250 + k, width=1)
    k += j
    j += 0.3

mainloop()
```

以上实例输出结果为：



Python 练习实例 57



题目：画图，学用 line 画直线。

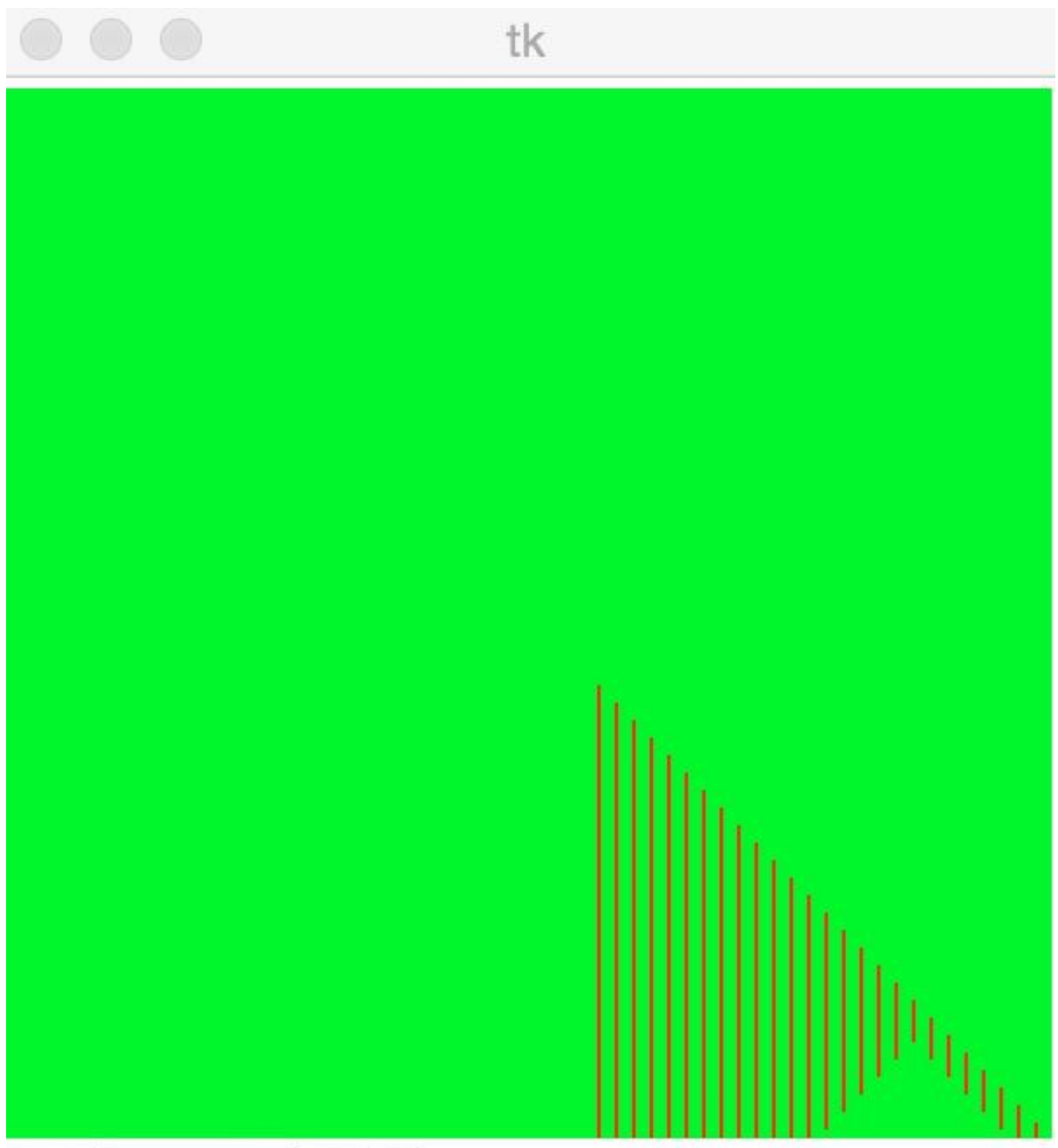
程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
```

```
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()
```

以上实例输出结果为：



Python 练习实例 58

 [Python 100 例](#)

题目：画图，学用 `rectangle` 画方形。

程序分析：

```
rectangle(int left, int top, int right, int bottom)
```

参数说明: (left , top)为矩形的左上坐标, (right,bottom)为矩形的右下坐标, 两者可确定一个矩形的大小

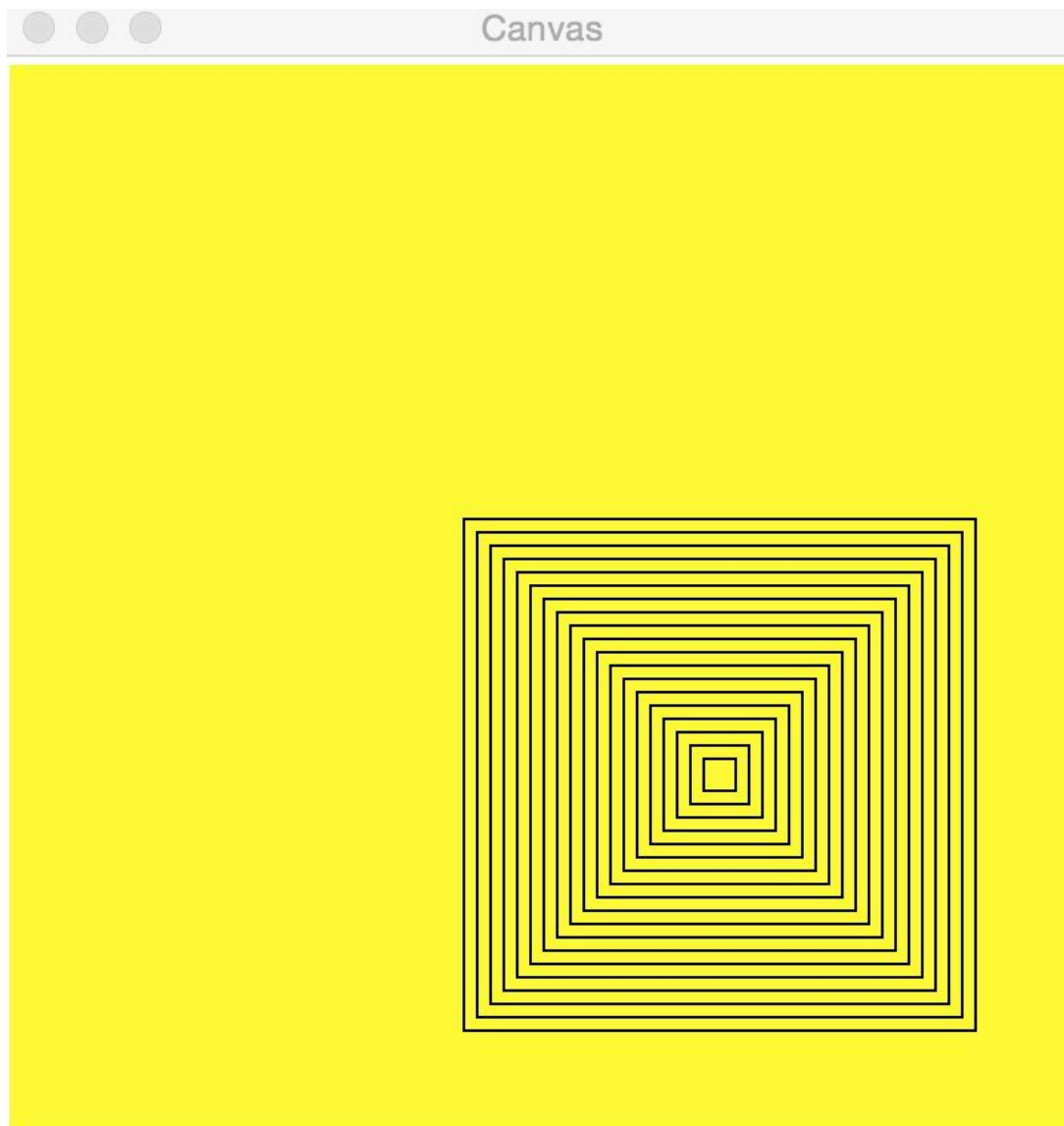
程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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()
```

以上实例输出结果为:



Python 练习实例 59



[Python 100 例](#)

题目：画图，综合例子。

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。。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    from Tkinter import *
    canvas = Canvas(width = 300,height = 300,bg = 'green')
```

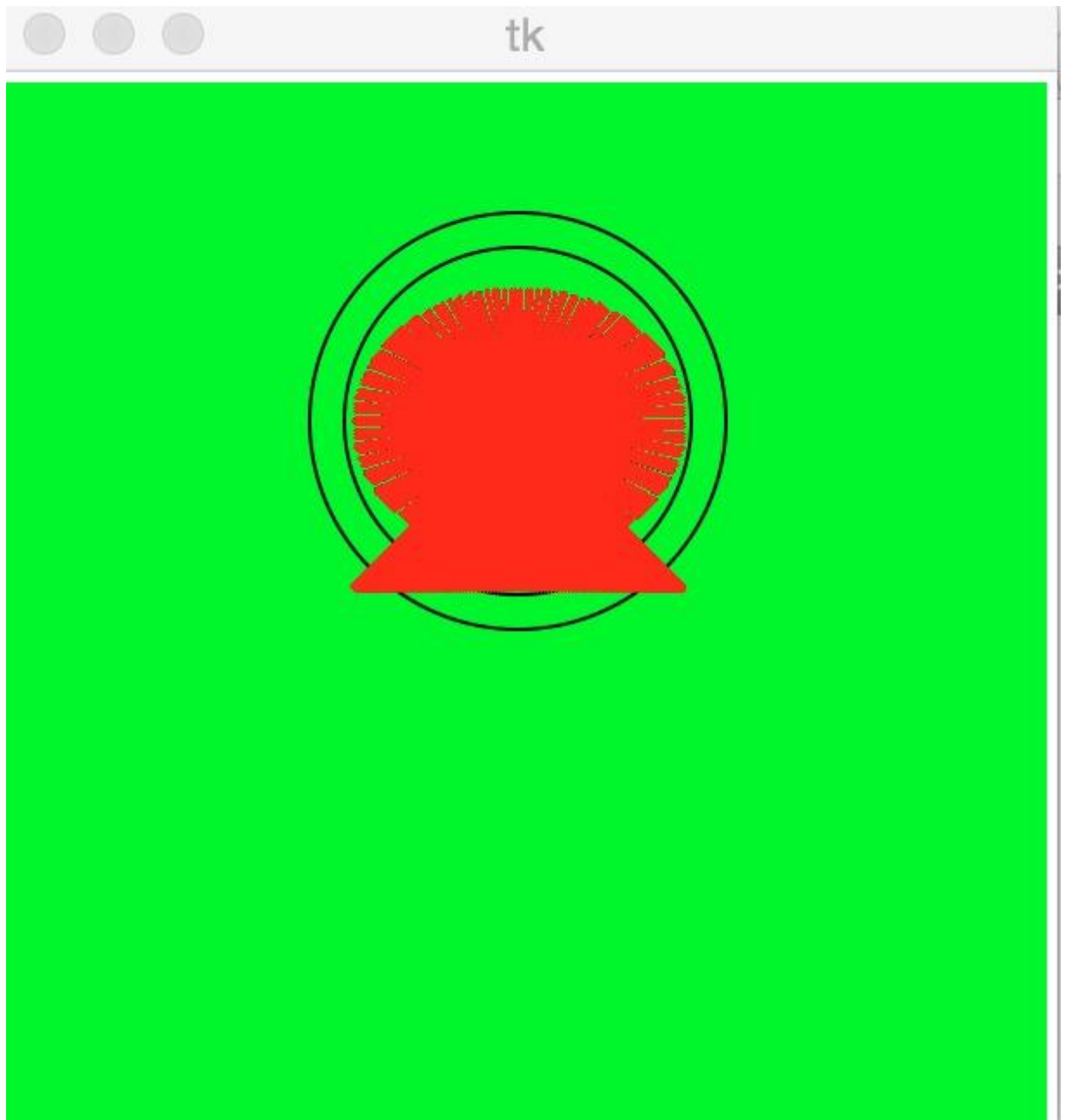
```

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()

```

以上实例输出结果为：



Python 练习实例 60

 [Python 100 例](#)

题目：计算字符串长度。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
```

```
sStr1 = 'strlen'
print len(sStr1)
```

以上实例输出结果为：

6

Python 练习实例 61



题目：打印出杨辉三角形（要求打印出 10 行如下图）。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = []
    for i in range(10):
        a.append([])
        for j in range(10):
            a[i].append(0)
    for i in range(10):
        a[i][0] = 1
        a[i][i] = 1
    for i in range(2,10):
        for j in range(1,i):
            a[i][j] = a[i - 1][j-1] + a[i - 1][j]
    from sys import stdout
    for i in range(10):
        for j in range(i + 1):
            stdout.write(str(a[i][j]))
            stdout.write(' ')
        print
```

以上实例输出结果为：

```
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
```

Python 练习实例 62

 [Python 100 例](#)

题目：查找字符串。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

sStr1 = 'ABCDEFGH'
sStr2 = 'cde'
print sStr1.find(sStr2)
```

以上实例输出结果为：

2

Python 练习实例 63

 [Python 100 例](#)

题目：画椭圆 ellipse。

程序分析：无。

程序源代码：

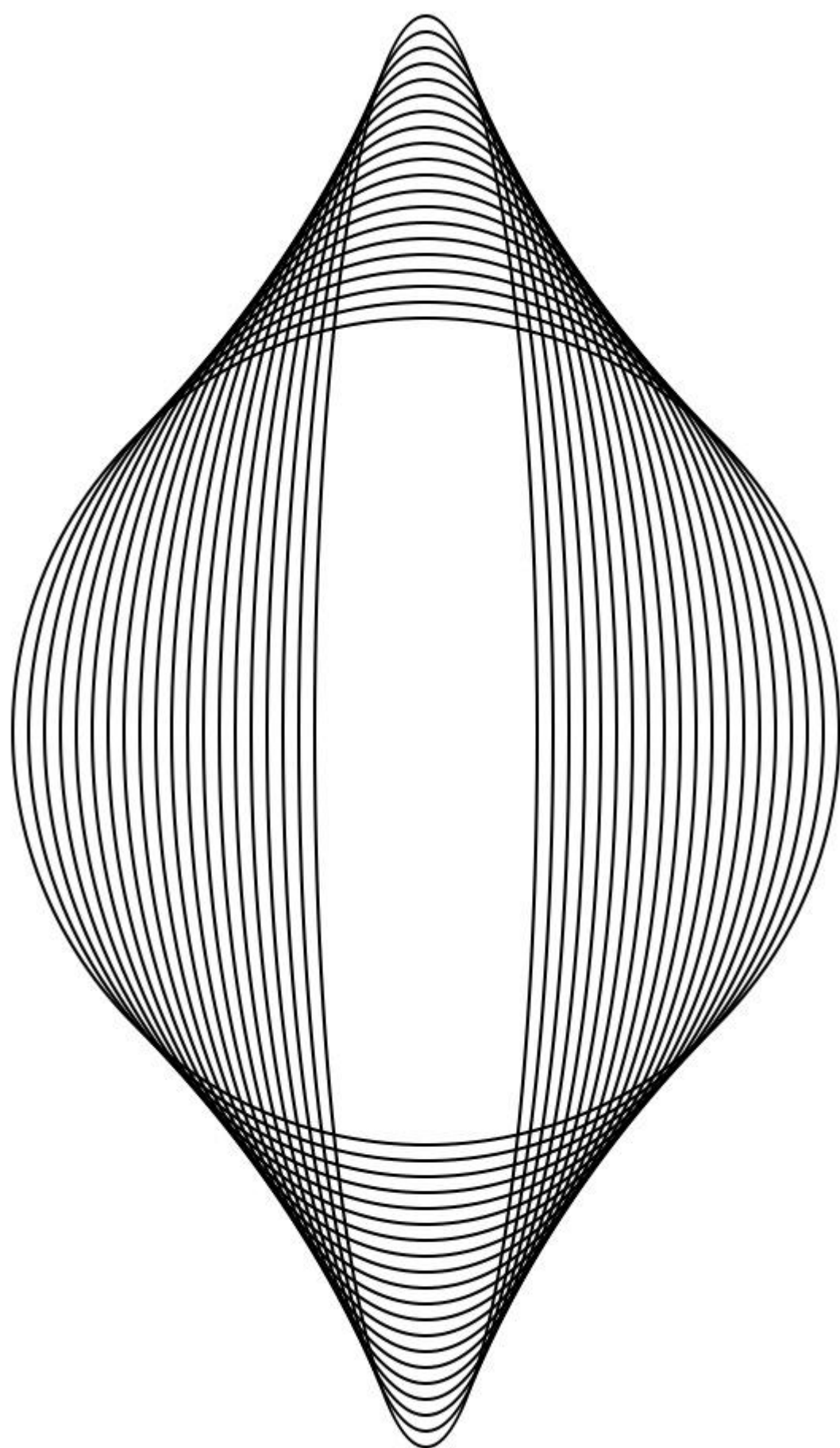
```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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()
```

以上实例输出结果为：



Python 练习实例 64



题目：利用 ellipse 和 rectangle 画图。。

程序分析：无。

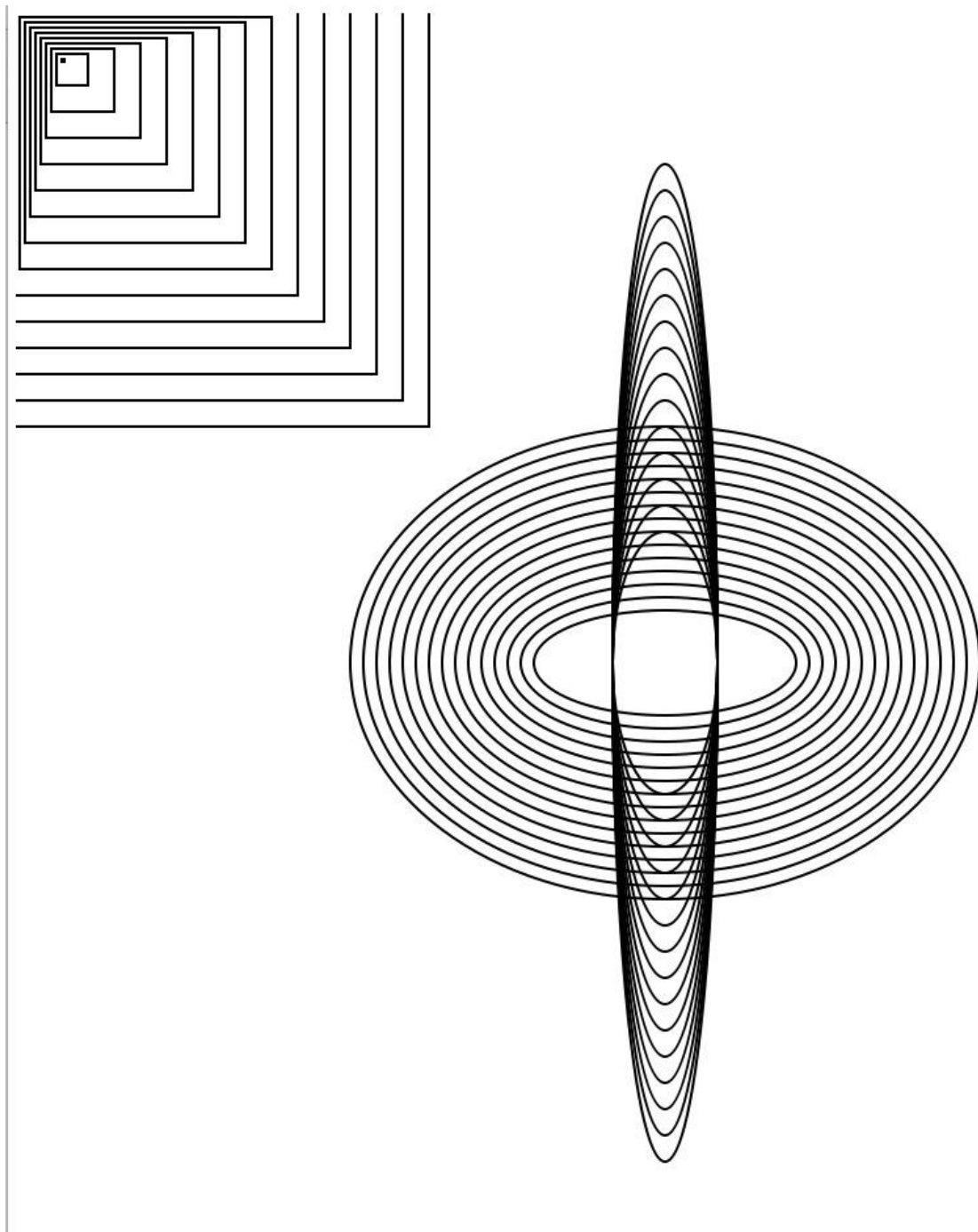
程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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()
```

以上实例输出结果为：



Python 练习实例 65

 [Python 100 例](#)

题目：一个最优美的图案。

程序分析：无。

程序源代码：

```

#!/usr/bin/python
# -*- coding: UTF-8 -*-

import math
class PTS:
    def __init__(self):
        self.x = 0
        self.y = 0
points = []

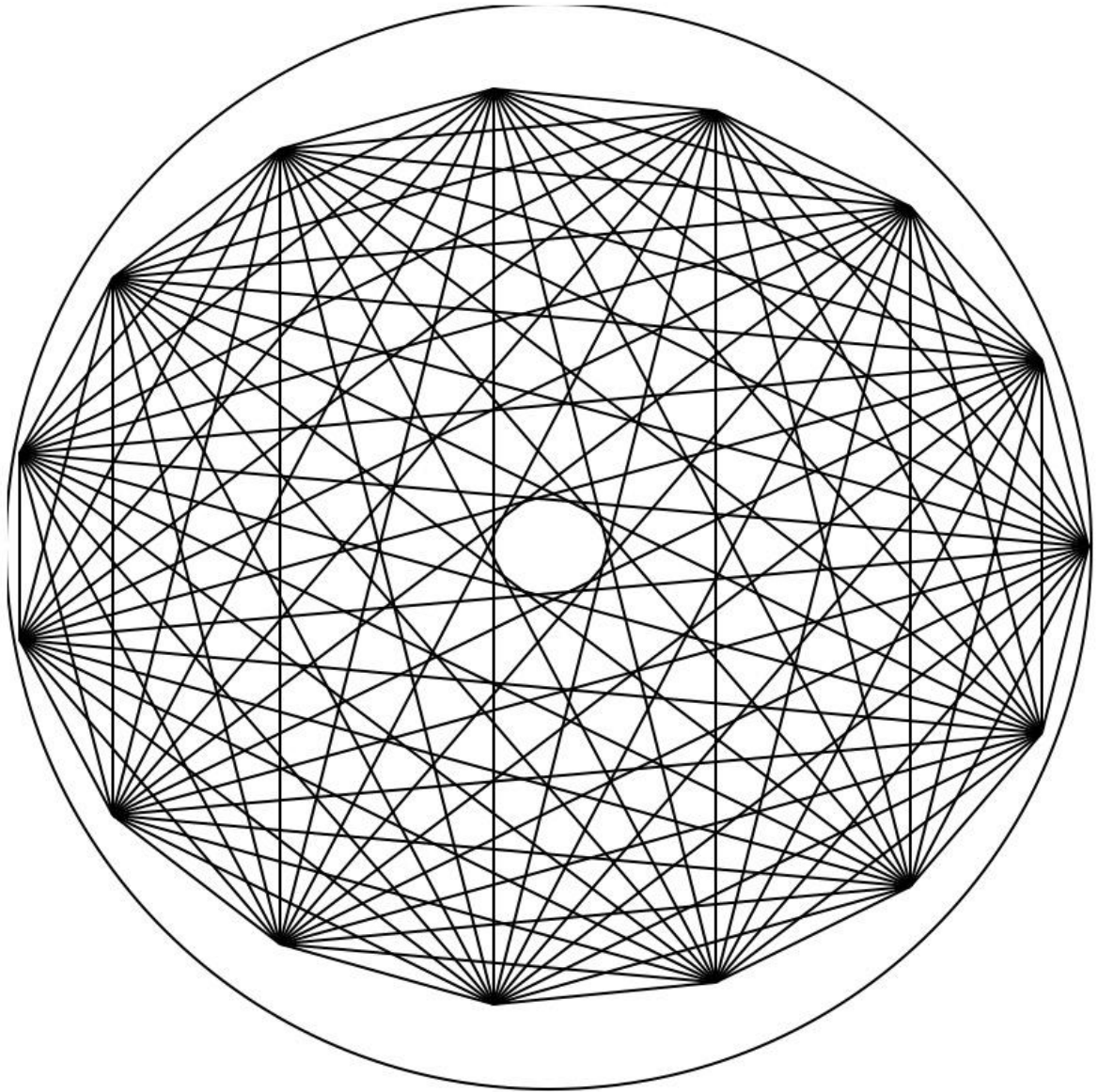
def LineToDemo():
    from Tkinter import *
    screenx = 400
    screeny = 400
    canvas = Canvas(width = screenx,height = screeny,bg = 'white')

    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()

```

以上实例输出结果为：



Python 练习实例 66

 [Python 100 例](#)

题目：输入 3 个数 a,b,c，按大小顺序输出。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
```

```

if __name__ == '__main__':
    n1 = int(raw_input('n1 = :\n'))
    n2 = int(raw_input('n2 = :\n'))
    n3 = int(raw_input('n3 = :\n'))

    def swap(p1,p2):
        return p2,p1

    if n1 > n2 : n1,n2 = swap(n1,n2)
    if n1 > n3 : n1,n3 = swap(n1,n3)
    if n2 > n3 : n2,n3 = swap(n2,n3)

    print n1,n2,n3

```

以上实例输出结果为：

```

n1 = :
123
n2 = :
456
n3 = :
789
123 456 789

```

Python 练习实例 67



[Python 100 例](#)

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。

程序源代码：

```

#!/usr/bin/python
# -*- coding: UTF-8 -*-

def inp(numbers):
    for i in range(9):
        numbers.append(int(raw_input('input a number:\n')))
    numbers.append(int(raw_input('input a number:\n')))
p = 0
def max_min(array):
    max = min = 0

```

```

    for i in range(1, len(array) - 1):
        p = i
        if array[p] > array[max] : max = p
        elif array[p] < array[min] : min = p
    k = max
    l = min
    array[0], array[l] = array[l], array[0]
    array[9], array[k] = array[k], array[9]

def outp(numbers):
    for i in range(len(numbers)):
        print numbers[i]

if __name__ == '__main__':
    array = []
    inp(array)
    max_min(array)
    outp(array)

```

以上实例输出结果为:

```

input a number:
6
input a number:
8
input a number:
3
input a number:
6
input a number:
4
input a number:
9
input a number:
334
input a number:
34
input a number:
23
input a number:
12
3
8
6

```

```
6
4
9
12
34
23
334
```

Python 练习实例 68



Python 100 例

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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    n = int(raw_input('the total number is:\n'))
    m = int(raw_input('back m:\n'))

    def move(array,n,m):
        array_end = array[n - 1]
        for i in range(n - 1,-1,- 1):
            array[i] = array[i - 1]
        array[0] = array_end
        m -= 1
        if m > 0:move(array,n,m)

    number = []
    for i in range(n):
        number.append(int(raw_input('input a number:\n')))
    print 'original number:',number

    move(number,n,m)

    print 'after moved:',number
```

以上实例输出结果为：

```

the total number is:
8
back m:
5
input a number:
2
input a number:
8
input a number:
6
input a number:
1
input a number:
78
input a number:
45
input a number:
34
input a number:
2
original number: [2, 8, 6, 1, 78, 45, 34, 2]
after moved: [1, 78, 45, 34, 2, 2, 8, 6]

```

Python 练习实例 69



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。

程序源代码：

```

#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    nmax = 50
    n = int(raw_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]

```

执行以上代码，输出结果：

```

$ python test.py
请输入总人数:34
10

```

Python 练习实例 70



[Python 100 例](#)

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。

程序源代码：

```

#!/usr/bin/python

# -*- coding: UTF-8 -*-

if __name__ == '__main__':

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

```

```
print 'the string has %d characters.' % len(s)
```

以上实例输出结果为：

```
please input a string:
```

```
abc
```

```
the string has 3 characters.
```

Python 练习实例 71



[Python 100 例](#)

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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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] = raw_input('input student num:\n')
        stu[i][1] = raw_input('input student name:\n')
        for j in range(3):
            stu[i][2].append(int(raw_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)

```

以上实例输出结果为：

```

input student num:
2
input student name:
aaa
score:
89
score:
98
score:
67
input student num:
bbb
input student name:
ccc
score:
87
score:
45
score:
68
input student num:
ddd
input student name:
eee
score:
56
score:
78
score:
56
[['2', 'aaa', [89, 98, 67]], ['bbb', 'ccc', [87, 45, 68]], ['ddd', 'eee', [56,
78, 56]], ['', '', []], ['', '', []]]
2      aaa
89

```

```
98
67
bbb   ccc
87
45
68
ddd   eee
56
78
56
```

Python 练习实例 72



题目：创建一个链表。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    ptr = []
    for i in range(5):
        num = int(raw_input('please input a number:\n'))
        ptr.append(num)
    print ptr
```

以上实例输出结果为：

```
please input a number:
3
please input a number:
5
please input a number:
7
please input a number:
8
please input a number:
2
```

```
[3, 5, 7, 8, 2]
```

Python 练习实例 73

 [Python 100 例](#)

题目：反向输出一个链表。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    ptr = []
    for i in range(5):
        num = int(raw_input('please input a number:\n'))
        ptr.append(num)
    print ptr
    ptr.reverse()
    print ptr
```

以上实例输出结果为：

```
please input a number:
6
please input a number:
5
please input a number:
3
please input a number:
4
please input a number:
8
[6, 5, 3, 4, 8]
[8, 4, 3, 5, 6]
```

Python 练习实例 74

 [Python 100 例](#)

题目：连接两个链表。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    arr1 = (3,12,8,9,11)
    ptr = list(arr1)
    print ptr
    ptr.sort()
    print ptr
[3, 12, 8, 9, 11]
[3, 8, 9, 11, 12]
```

Python 练习实例 75



Python 100 例

题目：放松一下，算一道简单的题目。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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
```

以上实例输出结果为：

67

Python 练习实例 76



题目：编写一个函数，输入 n 为偶数时，调用函数求 $1/2+1/4+\dots+1/n$,当输入 n 为奇数时，调用函数 $1/1+1/3+\dots+1/n$ (利用指针函数)

程序分析：无。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

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 / i

    return s

def dcall(fp,n):

    s = fp(n)

    return s
```

```
if __name__ == '__main__':  
  
    n = int(raw_input('input a number:\n'))  
  
    if n % 2 == 0:  
  
        sum = dcall(peven,n)  
  
    else:  
  
        sum = dcall(podd,n)  
  
    print sum
```

以上实例输出结果为:

```
input a number:  
  
6  
  
0.916666666667
```

Python 练习实例 77



题目：循环输出列表

程序分析：无。

程序源代码：

```
#!/usr/bin/python  
# -*- coding: UTF-8 -*-  
  
if __name__ == '__main__':  
    s = ["man","woman","girl","boy","sister"]  
    for i in range(len(s)):  
        print s[i]
```

以上实例输出结果为:

```
man
```

```
woman
girl
boy
sister
```

Python 练习实例 78



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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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])
```

以上实例输出结果为：

```
wang,50
```

Python 练习实例 79



题目：字符串排序。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
```

```

if __name__ == '__main__':
    str1 = raw_input('input string:\n')
    str2 = raw_input('input string:\n')
    str3 = raw_input('input string:\n')
    print str1,str2,str3

    if str1 > str2 : str1,str2 = str2,str1
    if str1 > str3 : str1,str3 = str3,str1
    if str2 > str3 : str2,str3 = str3,str2

    print 'after being sorted.'
    print str1,str2,str3

```

以上实例输出结果为：

```

input string:
abcde
input string:
efdis
input string:
adk
abcde efdis adk
after being sorted.
abcde adk efdis

```

Python 练习实例 80



[Python 100 例](#)

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。

程序源代码：

```

#!/usr/bin/python

# -*- coding: UTF-8 -*-

```

```
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
```

以上实例输出结果为：

3121

Python 练习实例 81



题目：809*??=800*??+9*??+1 其中??代表的两位数,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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

a = 809
for i in range(10,100):
    b = i * a + 1
    if b >= 1000 and b <= 10000 and 8 * i < 100 and 9 * i >= 100:
        print b, '/', i, ' = 809 * ', i, ' + ', b % i
```

以上实例输出结果为：

```
9709 / 12 = 809 * 12 + 1
```

Python 练习实例 82

 [Python 100 例](#)

题目：八进制转换为十进制

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    n = 0
    p = raw_input('input a octal number:\n')
    for i in range(len(p)):
        n = n * 8 + ord(p[i]) - ord('0')
    print n
```

以上实例输出结果为：

```
input a octal number:
f
54
```

Python 练习实例 83

 [Python 100 例](#)

题目：求 0—7 所能组成的奇数个数。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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
```

以上实例输出结果为：

```
4
32
256
2048
16384
131072
1048576
sum = 8388608
```

Python 练习实例 84

 [Python 100 例](#)

题目：连接字符串。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
```

```
delimiter = ','  
mylist = ['Brazil', 'Russia', 'India', 'China']  
print delimiter.join(mylist)
```

以上实例输出结果为:

```
Brazil,Russia,India,China
```

Python 练习实例 85



[Python 100 例](#)

题目：判断一个素数能被几个 9 整除。

程序分析：无。

程序源代码：

```
#!/usr/bin/python  
# -*- coding: UTF-8 -*-  
  
if __name__ == '__main__':  
    zi = int(raw_input('input a number:\n'))  
    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 can be divided by %d 9' % (sum,c9)
```

以上实例输出结果为:

```
input a number:  
9  
9 can be divided by 1 9
```

Python 练习实例 86

Python 100 例

题目：两个字符串连接程序。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    a = "acegikm"
    b = "bdfhjlnpq"

    # 连接字符串
    c = a + b

    print c
```

以上实例输出结果为：

```
acegikmbdfhjlnpq
```

Python 练习实例 87

Python 100 例

题目：回答结果（结构体变量传递）。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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
```

以上实例输出结果为：

```
20 c
```

Python 练习实例 88



[Python 100 例](#)

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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    n = 1
    while n <= 7:
        a = int(raw_input('input a number:\n'))
        while a < 1 or a > 50:
            a = int(raw_input('input a number:\n'))
        print a * '*'
        n += 1
```

以上实例输出结果为：

```
input a number:
9
*****
input a number:
5
*****
input a number:
6
*****
input a number:
```

Python 练习实例 89



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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from sys import stdout
if __name__ == '__main__':
    a = int(raw_input('input a number:\n'))
    aa = []
    aa.append(a % 10)
    aa.append(a % 100 / 10)
    aa.append(a % 1000 / 100)
    aa.append(a / 1000)

    for i in range(4):
        aa[i] += 5
        aa[i] %= 10
    for i in range(2):
        aa[i], aa[3 - i] = aa[3 - i], aa[i]
    for i in range(3, -1, -1):
        stdout.write(str(aa[i]))
```

以上实例输出结果为：

```
input a number:
89
4355
```

Python 练习实例 90



题目：列表使用实例。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

#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
```

以上实例输出结果为：

```
3
['\xe4\xb8\xad\xe5\x9b\xbd\xe7\xa7\xbb\xe5\x8a\xa8', [1, 2, 4, 5]]
4
i'm new here!
中国移动
3
```

```
[10086, [1, 2, 4, 5], "i'm new here!"]  
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
[4, 5, 6]  
[2, 5, 8]  
[2, 8]
```

Python 练习实例 91



题目：时间函数举例 1。

程序分析：无。

程序源代码：

```
#!/usr/bin/python  
  
# -*- coding: UTF-8 -*-  
  
if __name__ == '__main__':  
  
    import time  
  
    print time.ctime(time.time())  
  
    print time.asctime(time.localtime(time.time()))  
  
    print time.asctime(time.gmtime(time.time()))
```

以上实例输出结果为：

```
Wed Oct 21 17:08:51 2015  
  
Wed Oct 21 17:08:51 2015  
  
Wed Oct 21 09:08:51 2015
```

Python 练习实例 92

Python 100 例

题目：时间函数举例 2。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    import time
    start = time.time()
    for i in range(3000):
        print i
    end = time.time()

    print end - start
```

以上实例输出结果为：

```
0
1
2
3
4
.....
```

Python 练习实例 93

Python 100 例

题目：时间函数举例 3。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    import time
    start = time.clock()
```

```
for i in range(10000):
    print i
end = time.clock()
print 'different is %6.3f' % (end - start)
```

以上实例输出结果为：

```
0
1
2
3
4
.....
different is 0.014
```

Python 练习实例 94



[Python 100 例](#)

题目：时间函数举例 4,一个猜数游戏，判断一个人反应快慢。

程序分析：无。

程序源代码：

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-

if __name__ == '__main__':

    import time

    import random

    play_it = raw_input('do you want to play it.(\n'y\n' or \n\n)')

    while play_it == 'y':

        c = raw_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(raw_input('input your guess:\n'))

while guess != i:

    if guess > i:

        print 'please input a little smaller'

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

    else:

        print 'please input a little bigger'

        guess = int(raw_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 = raw_input('do you want to play it.')
```

以上实例输出结果为：

```
do you want to play it.('y' or 'n')y
input a character:
5
please input number you guess:

input your guess:
60
.....
please input a little bigger
input your guess:
29
please input a little smaller
input your guess:
28
3.81868131868e-05
you are very clever!
Congradulations
The number you guess is 28
do you want to play it.
```

Python 练习实例 95

 [Python 100 例](#)

题目：字符串日期转换为易读的日期格式。

程序分析：无。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from dateutil import parser
dt = parser.parse("Aug 28 2015 12:00AM")
print dt
```

以上实例输出结果为:

```
2015-08-28 00:00:00
```

Python 练习实例 96



[Python 100 例](#)

题目: 计算字符串中子串出现的次数。

程序分析: 无。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    str1 = raw_input('input a string:\n')
    str2 = raw_input('input a sub string:\n')
    ncount = str1.count(str2)
    print ncount
```

以上实例输出结果为:

```
input a string:
www.runoob.com
input a sub string:
runoob
1
```

Python 练习实例 97



[Python 100 例](#)

题目：从键盘输入一些字符，逐个把它们送到磁盘上去，直到输入一个#为止。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
    from sys import stdout
    filename = raw_input('input a file name:\n')
    fp = open(filename, "w")
    ch = raw_input('input string:\n')
    while ch != '#':
        fp.write(ch)
        stdout.write(ch)
        ch = raw_input('')
    fp.close()
```

以上实例输出结果为：

```
input a file name:
a
input string:
b
b
c
c#
```

Python 练习实例 98



[Python 100 例](#)

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。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

if __name__ == '__main__':
```

```
fp = open('test.txt','w')
string = raw_input('please input a string:\n')
string = string.upper()
fp.write(string)
fp = open('test.txt','r')
print fp.read()
fp.close()
```

以上实例输出结果为:

```
please input a string:
runoob.com
RUNOOB.COM
```

Python 练习实例 99



[Python 100 例](#)

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。

程序源代码:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

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()
```

运行以上程序前，你需要在脚本执行的目录下创建 test1.txt、test2.txt 文件。

以上程序执行成功后，打开 test3.txt 文件可以看到内容如下所示：

```
123456
```

Python 练习实例 100



题目：列表转换为字典。

程序分析：无。

程序源代码：

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

i = ['a', 'b']
l = [1, 2]
print dict([i,l])
```

以上实例输出结果为：

```
{'a': 'b', 1: 2}
```