

# Python3 100例

---



## 实例001：数字组合

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

**\*\*程序分析：\*\***遍历全部可能，把有重复的剃掉。

```
1 total=0
2 for i in range(1,5):
3     for j in range(1,5):
4         for k in range(1,5):
5             if ((i!=j)and(j!=k)and(k!=i)):
6                 print(i,j,k)
7                 total+=1
8 print(total)
```

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

```
1 import itertools
2 sum2=0
3 a=[1,2,3,4]
4 for i in itertools.permutations(a,3):
5     print(i)
6     sum2+=1
7 print(sum2)
```

## 实例002：“个税计算”

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

**\*\*程序分析：**\*\*分区间计算即可。

```
1 profit=int(input('Show me the money: '))
2 bonus=0
3 thresholds=[100000,100000,200000,200000,400000]
4 rates=[0.1,0.075,0.05,0.03,0.015,0.01]
5 for i in range(len(thresholds)):
6     if profit<=thresholds[i]:
7         bonus+=profit*rates[i]
8         profit=0
9         break
10    else:
11        bonus+=thresholds[i]*rates[i]
12        profit-=thresholds[i]
13 bonus+=profit*rates[-1]
14 print(bonus)
```

### 实例003：完全平方数

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

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

```
1 n=0
2 while (n+1)**2-n*n<=168:
3     n+=1
4
5 print(n+1)
6 -----
7
8 85
```

思路是：最坏的结果是n的平方与(n+1)的平方刚好差168，由于是平方的关系，不可能存在比这更大的间隙。至于判断是否是完全平方数，最简单的方法是：平方根的值小数部分为0即可。结合起来：

```
1 n=0
2 while (n+1)**2-n*n<=168:
3     n+=1
4
5 for i in range((n+1)**2):
6     if i**0.5==int(i**0.5) and (i+168)**0.5==int((i+168)**0.5):
7         print(i-100)
```

### 实例004：这天第几天

**\*\*题目：**\*\*输入某年某月某日，判断这一天是这一年的第几天？

**\*\*程序分析：**\*\*特殊情况，闰年时需考虑二月多加一天：

```

1 def isLeapYear(y):
2     return (y%4==0 or (y%4==0 and y%100!=0))
3 DofM=[0,31,28,31,30,31,30,31,31,30,31,30]
4 res=0
5 year=int(input('Year:'))
6 month=int(input('Month:'))
7 day=int(input('day:'))
8 if isLeapYear(year):
9     DofM[2]+=1
10 for i in range(month):
11     res+=DofM[i]
12 print(res+day)

```

### 实例005：三数排序

**\*\*题目：**\*\*输入三个整数x,y,z，请把这三个数由小到大输出。

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

```

1 raw=[]
2 for i in range(3):
3     x=int(input('int%d:'%(i)))
4     raw.append(x)
5
6 for i in range(len(raw)):
7     for j in range(i+1,len(raw)):
8         if raw[i]>raw[j]:
9             raw[i],raw[j]=raw[j],raw[i]
10 print(raw)
11
12
13 raw2=[]
14 for i in range(3):
15     x=int(input('int%d:'%(i)))
16     raw2.append(x)
17 print(sorted(raw2))

```

### 实例006：斐波那契数列

**\*\*题目：**\*\*斐波那契数列。

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

```

1 # 递归实现
2 def Fib(n):
3     return 1 if n<=2 else Fib(n-1)+Fib(n-2)
4 print(Fib(int(input())))
5
6 # 朴素实现

```

```

7 target=int(input())
8 res=0
9 a,b=1,1
10 for i in range(target-1):
11     a,b=b,a+b
12 print(a)

```

### 实例007: copy

**\*\*题目:** \*\*将一个列表的数据复制到另一个列表中。

**\*\*程序分析:** \*\*使用列表[:], 拿不准可以调用copy模块。

```

1 import copy
2 a = [1,2,3,4,['a','b']]
3
4 b = a                # 赋值
5 c = a[:]             # 浅拷贝
6 d = copy.copy(a)     # 浅拷贝
7 e = copy.deepcopy(a) # 深拷贝
8
9 a.append(5)
10 a[4].append('c')
11
12 print('a=',a)
13 print('b=',b)
14 print('c=',c)
15 print('d=',d)
16 print('e=',e)
17
18 ===== RESTART: F:\PyWorkspace\Python100\100examples\007.py
19 =====
19 a= [1, 2, 3, 4, ['a', 'b', 'c'], 5]
20 b= [1, 2, 3, 4, ['a', 'b', 'c'], 5]
21 c= [1, 2, 3, 4, ['a', 'b', 'c']]
22 d= [1, 2, 3, 4, ['a', 'b', 'c']]
23 e= [1, 2, 3, 4, ['a', 'b']]

```

### 实例008:九九乘法表

**\*\*题目:** \*\*输出 9\*9 乘法口诀表。

**\*\*程序分析:** \*\*分行与列考虑, 共9行9列, i控制行, j控制列。

```

1 for i in range(1,10):
2     for j in range(1,i+1):
3         print('%d*%d=%2ld'%(i,j,i*j),end='')
4     print()

```

### 实例009: 暂停一秒输出

**\*\*题目：**\*\*暂停一秒输出。

**\*\*程序分析：**\*\*使用 time 模块的 sleep() 函数。

```
1 import time
2 for i in range(4):
3     print(str(int(time.time()))[-2:])
4     time.sleep(1)
```

### 实例010：给人看的时间

**\*\*题目：**\*\*暂停一秒输出，并格式化当前时间。

**\*\*程序分析：**\*\*同009。

```
1 import time
2
3 for i in range(4):
4     print(time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time())))
5     time.sleep(1)
```

### 实例011：养兔子

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

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

```
1 month=int(input('繁殖几个月? : '))
2 month_1=1
3 month_2=0
4 month_3=0
5 month_elder=0
6 for i in range(month):
7
8     month_1,month_2,month_3,month_elder=month_elder+month_3,month_1,month_2,month_elder+month_3
9     print('第%d个月共'%(i+1),month_1+month_2+month_3+month_elder,'对兔子')
10    print('其中1月兔: ',month_1)
11    print('其中2月兔: ',month_2)
12    print('其中3月兔: ',month_3)
13    print('其中成年兔: ',month_elder)
```

### 实例012:100到200的素数

**\*\*题目：**\*\*判断101–200之间有多少个素数，并输出所有素数。

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

```

1 import math
2 for i in range(100,200):
3     flag=0
4     for j in range(2,round(math.sqrt(i))+1):
5         if i%j==0:
6             flag=1
7             break
8     if flag:
9         continue
10    print(i)
11
12
13 print('\nSimplify the code with "else"\n')
14
15
16 for i in range(100,200):
17     for j in range(2,round(math.sqrt(i))+1):
18         if i%j==0:
19             break
20     else:
21         print(i)

```

### 实例013：所有水仙花数

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

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

```

1 for i in range(100,1000):
2     s=str(i)
3     one=int(s[-1])
4     ten=int(s[-2])
5     hun=int(s[-3])
6     if i == one**3+ten**3+hun**3:
7         print(i)

```

### 实例014：分解质因数

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

**\*\*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。

```

1 target=int(input('输入一个整数: '))
2 print(target,'= ',end='')
3
4 if target<0:
5     target=abs(target)

```

```

6     print('-1*',end='')
7
8     flag=0
9     if target<=1:
10         print(target)
11         flag=1
12
13
14     while True:
15         if flag:
16             break
17         for i in range(2,int(target+1)):
18             if target%i==0:
19                 print("%d"%i,end='')
20                 if target==i:
21                     flag=1
22                     break
23                 print('*',end='')
24                 target/=i
25                 break
26

```

### 实例015：分数归档

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

**\*\*程序分析：**\*\*用条件判断即可。

```

1 points=int(input('输入分数: '))
2 if points>=90:
3     grade='A'
4 elif points<60:
5     grade='C'
6 else:
7     grade='B'
8 print(grade)

```

### 实例016：输出日期

**\*\*题目：**\*\*输出指定格式的日期。

**\*\*程序分析：**\*\*使用 datetime 模块。

```

1 import datetime
2 print(datetime.date.today())
3 print(datetime.date(2333,2,3))
4 print(datetime.date.today().strftime('%d/%m/%Y'))
5 day=datetime.date(1111,2,3)
6 day=day.replace(year=day.year+22)
7 print(day)

```

### 实例017：字符串构成

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

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

```
1 string=input("输入字符串: ")
2 alp=0
3 num=0
4 spa=0
5 oth=0
6 for i in range(len(string)):
7     if string[i].isspace():
8         spa+=1
9     elif string[i].isdigit():
10        num+=1
11    elif string[i].isalpha():
12        alp+=1
13    else:
14        oth+=1
15 print('space: ',spa)
16 print('digit: ',num)
17 print('alpha: ',alp)
18 print('other: ',oth)
```

### 实例018：复读机相加

**\*\*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)，几个数相加由键盘控制。

**\*\*程序分析：**\*\*用字符串解决。

```
1 a=input('被加数字: ')
2 n=int(input('加几次?: '))
3 res=0
4 for i in range(n):
5     res+=int(a)
6     a+=a[0]
7 print('结果是: ',res)
```

### 实例019：完数

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

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

```
1 def factor(num):
2     target=int(num)
3     res=set()
4     for i in range(1,num):
```



```

5         if num%i==0:
6             res.add(i)
7             res.add(num/i)
8     return res
9
10    for i in range(2,1001):
11        if i==sum(factor(i))-i:
12            print(i)

```

### 实例020：高空抛物

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

**\*\*程序分析：\*\***无

```

1    high=200.
2    total=100
3    for i in range(10):
4        high/=2
5        total+=high
6        print(high/2)
7    print('总长: ',total)

```

### 实例021：猴子偷桃

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

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

```

1    peach=1
2    for i in range(9):
3        peach=(peach+1)*2
4    print(peach)

```

### 实例022：比赛对手

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

**\*\*程序分析：\*\***找到条件下不重复的三个对手即可。

```

1    a=set(['x','y','z'])
2    b=set(['x','y','z'])
3    c=set(['x','y','z'])
4    c-=set(('x','z'))
5    a-=set('x')

```

```

6 for i in a:
7     for j in b:
8         for k in c:
9             if len(set((i,j,k)))==3:
10                 print('a:%s,b:%s,c:%s'%(i,j,k))

```

### 实例023：画菱形

**\*\*题目：**\*\*打印出如下图案（菱形）：

```

    *   ***   *****   *****   *****   ***   *

```

**\*\*程序分析：**\*\*递归调用即可。

```

1 def draw(num):
2     a="*(2*(4-num)+1)
3     print(a.center(9,' '))
4     if num!=1:
5         draw(num-1)
6         print(a.center(9,' '))
7 draw(4)

```

### 实例024：斐波那契数列II

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

**\*\*程序分析：**\*\*就是斐波那契数列的后一项除以前一项。

```

1 a = 2.0
2 b = 1.0
3 s = 0
4 for n in range(1,21):
5     s += a / b
6     a,b = a + b,a
7 print (s)

```

### 实例025：阶乘求和

**\*\*题目：**\*\*求1+2!+3!+...+20!的和。

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

```

1 res=1
2 for i in range(20,1,-1):
3     res=i*res+1
4 print(res)

```

### 实例026：递归求阶乘

**\*\*题目：**\*\*利用递归方法求5!。

**\*\*程序分析：**\*\*递归调用即可。

```
1 def factorial(n):
2     return n*factorial(n-1) if n>1 else 1
3 print(factorial(5))
```

### 实例027：递归输出

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

**\*\*程序分析：**\*\*递归真是蠢方法。

```
1 def rec(string):
2     if len(string)!=1:
3         rec(string[1:])
4     print(string[0],end='')
5
6 rec(input('string here:'))
```

### 实例028：递归求等差数列

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

**\*\*程序分析：**\*\*就一等差数列。

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

### 实例029：反向输出

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

**\*\*程序分析：**\*\*学会分解出每一位数,用字符串的方法总是比较省事。

```
1 n=int(input('输入一个正整数: '))
2 n=str(n)
3 print('%d位数'%len(n))
4 print(n[::-1])
```

### 实例030：回文数

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

**\*\*程序分析：**\*\*用字符串比较方便,就算输入的不是数字都ok。

```
1 n=input("随便你输入啥啦: ")
2 a=0
```

```

3  b=len(n)-1
4  flag=True
5  while a<b:
6      if n[a]!=n[b]:
7          print('不是回文串')
8          flag=False
9          break
10     a,b=a+1,b-1
11 if flag:
12     print('是回文串')

```

### 实例031：字母识词

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

**\*\*程序分析：**\*\*这里用字典的形式直接将对照关系存好。

```

1  weekT={'h':'thursday',
2         'u':'tuesday'}
3  weekS={'a':'saturday',
4         'u':'sunday'}
5  week={'t':weekT,
6        's':weekS,
7        'm':'monday',
8        'w':'wensday',
9        'f':'friday'}
10 a=week[str(input('请输入第一位字母:')).lower()]
11 if a==weekT or a==weekS:
12     print(a[str(input('请输入第二位字母:')).lower()])
13 else:
14     print(a)

```

### 实例032：反向输出II

**\*\*题目：**\*\*按相反的顺序输出列表的值。

**\*\*程序分析：**\*\*无。

```

1  a = ['one', 'two', 'three']
2  print(a[::-1])

```

### 实例033：列表转字符串

**\*\*题目：**\*\*按逗号分隔列表。

**\*\*程序分析：**\*\*无。

```

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

```

### 实例034：调用函数

**\*\*题目：**\*\*练习函数调用。

**\*\*程序分析：**\*\*无。

```
1 def hello():
2     print('Hello World!')
3 def helloAgain():
4     for i in range(2):
5         hello()
6
7 if __name__=='__main__':
8     helloAgain()
```

### 实例035：设置输出颜色

**\*\*题目：**\*\*文本颜色设置。

**\*\*程序分析：**\*\*无。

```
1 class bcolors:
2     HEADER = '\033[95m'
3     OKBLUE = '\033[94m'
4     OKGREEN = '\033[92m'
5     WARNING = '\033[93m'
6     FAIL = '\033[91m'
7     ENDC = '\033[0m'
8     BOLD = '\033[1m'
9     UNDERLINE = '\033[4m'
10 print(bcolors.WARNING + "警告的颜色字体?" + bcolors.ENDC))
```

### 实例036：算素数

**\*\*题目：**\*\*求100之内的素数。

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

```
1 lo=int(input('下限: '))
2 hi=int(input('上限: '))
3 for i in range(lo,hi+1):
4     if i > 1:
5         for j in range(2,i):
6             if (i % j) == 0:
7                 break
8         else:
9             print(i)
```

### 实例037：排序

**\*\*题目：**\*\*对10个数进行排序。

**\*\*程序分析：**\*\*同实例005。

```

1 raw=[]
2 for i in range(10):
3     x=int(input('int%d:'%(i)))
4     raw.append(x)
5
6 for i in range(len(raw)):
7     for j in range(i,len(raw)):
8         if raw[i]>raw[j]:
9             raw[i],raw[j]=raw[j],raw[i]
10 print(raw)
11

```

### 实例038：矩阵对角线之和

**\*\*题目：**\*\*求一个3\*3矩阵主对角线元素之和。

**\*\*程序分析：**\*\*无。

```

1 mat=[[1,2,3],
2      [3,4,5],
3      [4,5,6]
4      ]
5 res=0
6 for i in range(len(mat)):
7     res+=mat[i][i]
8 print(res)

```

### 实例039：有序列表插入元素

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

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

```

1 lis=[1,10,100,1000,10000,100000]
2 n=int(input('insert a number: '))
3 lis.append(n)
4 for i in range(len(lis)-1):
5     if lis[i]>=n:
6         for j in range(i,len(lis)):
7             lis[j],lis[j+1]=lis[j+1],lis[j]
8         break
9 print(lis)

```

### 实例040：逆序列表

**\*\*题目：**\*\*将一个数组逆序输出。

**\*\*程序分析：**\*\*依次交换位置，或者直接调用reverse方法。

```

1  lis=[1,10,100,1000,10000,100000]
2  for i in range(int(len(lis)/2)):
3      lis[i],lis[len(lis)-1-i]=lis[len(lis)-1-i],lis[i]
4  print('第一种实现: ')
5  print(lis)
6
7
8  lis=[1,10,100,1000,10000,100000]
9  print('第二种实现: ')
10 lis.reverse()
11 print(lis)

```

### 实例041：类的方法与变量

**\*\*题目：**\*\*模仿静态变量的用法。

**\*\*程序分析：**\*\*构造类，了解类的方法与变量。

```

1  def dummy():
2      i=0
3      print(i)
4      i+=1
5
6  class cls:
7      i=0
8      def dummy(self):
9          print(self.i)
10         self.i+=1
11
12 a=cls()
13 for i in range(50):
14     dummy()
15     a.dummy()

```

### 实例042：变量作用域

**\*\*题目：**\*\*学习使用auto定义变量的用法。

**\*\*程序分析：**\*\*python中的变量作用域。

```

1  i=0
2  n=0
3  def dummy():
4      i=0
5      print(i)
6      i+=1
7  def dummy2():
8      global n
9      print(n)
10     n+=1

```

```

11 print('函数内部的同名变量')
12 for j in range(20):
13     print(i)
14     dummy()
15     i+=1
16 print('global声明同名变量')
17 for k in range(20):
18     print(n)
19     dummy2()
20     n+=10

```

实例043：作用域、类的方法与变量

**\*\*题目：**\*\*模仿静态变量(static)另一案例。

**\*\*程序分析：**\*\*综合实例041和实例042。

```

1 class dummy:
2     num=1
3     def Num(self):
4         print('class dummy num:',self.num)
5         print('global num: ',num)
6         self.num+=1
7
8 n=dummy()
9 num=1
10 for i in range(5):
11     num*=10
12     n.Num()

```

## 实例044：矩阵相加

**\*\*题目：**\*\*计算两个矩阵相加。

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

```

1 X = [[12,7,3],
2      [4 ,5,6],
3      [7 ,8,9]]
4
5 Y = [[5,8,1],
6      [6,7,3],
7      [4,5,9]]
8
9 res=[[0,0,0],
10      [0,0,0],
11      [0,0,0]]
12 for i in range(len(res)):
13     for j in range(len(res[0])):
14         res[i][j]=X[i][j]+Y[i][j]
15 print(res)

```



### 实例045：求和

**\*\*题目：**\*\*统计 1 到 100 之和。

**\*\*程序分析：**\*\*无

```
1 res=0
2 for i in range(1,101):
3     res+=i
4 print(res)
```

### 实例046：打破循环

**\*\*题目：**\*\*求输入数字的平方，如果平方运算后小于 50 则退出。

**\*\*程序分析：**\*\*无

```
1 while True:
2     try:
3         n=float(input('输入一个数字: '))
4     except:
5         print('输入错误')
6         continue
7     dn=n**2
8     print('其平方为: ',dn)
9     if dn<50:
10        print('平方小于50, 退出')
11        break
```

### 实例047：函数交换变量

**\*\*题目：**\*\*两个变量值用函数互换。

**\*\*程序分析：**\*\*无

```
1 def exc(a,b):
2     return (b,a)
3 a=0
4 b=10
5 a,b=exc(a,b)
6 print(a,b)
```

### 实例048：数字比大小

**\*\*题目：**\*\*数字比较。

**\*\*程序分析：**\*\*无

```
1 a=int(input('a='))
2 b=int(input('b='))
3 if a<b:
```

```
4     print('a<b')
5 elif a>b:
6     print('a>b')
7 else:
8     print('a=b')
```

### 实例049：lambda

**\*\*题目：**\*\*使用lambda来创建匿名函数。

**\*\*程序分析：**\*\*无

```
1 Max=lambda x,y:x*(x>=y)+y*(y>x)
2 Min=lambda x,y:x*(x<=y)+y*(y<x)
3
4 a=int(input('1:'))
5 b=int(input('2:'))
6
7 print(Max(a,b))
8 print(Min(a,b))
```

### 实例050：随机数

**\*\*题目：**\*\*输出一个随机数。

**\*\*程序分析：**\*\*使用 random 模块。

```
1 import random
2 print(random.uniform(10,20))
```

### 实例051：按位与

**\*\*题目：**\*\*学习使用按位与 & 。

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

```
1 a=0o77
2 print(a)
3 b=a&3
4 print(b)
5 b=b&7
6 print(b)
```

### 实例052：按位或

**\*\*题目：**\*\*学习使用按位或 | 。

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

```
1 a=0o77
```

```
2 print(a|3)
3 print(a|3|7)
```

### 实例053：按位异或

**\*\*题目：**\*\*学习使用按位异或 ^ 。

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

```
1 a=0o77
2 print(a^3)
3 print(a^3^7)
```

### 实例054：位取反、位移动

**\*\*题目：**\*\*取一个整数a从右端开始的4~7位。

**\*\*程序分析：**\*\*可以这样考虑： (1)先使a右移4位。 (2)设置一个低4位全为1,其余全为0的数。可用 $\sim(0<<4)$  (3)将上面二者进行&运算。

```
1 a=int(input('输入一个数字: '))
2 b=0 # 0
3 b=~b # 1
4 b=b<<4 # 10000
5 b=~b # 1111
6 c=a>>4
7 d=c&b
8 print('a:',bin(a))
9 print('b:',bin(b))
10 print('c:',bin(c))
11 print('d:',bin(d))
```

### 实例055：按位取反

**\*\*题目：**\*\*学习使用按位取反~。

**\*\*程序分析：**\*\* $\sim 0=1$ ;  $\sim 1=0$ ;

```
1 print(~234)
2 print(~~234)
```

### 实例056：画圈

**\*\*题目：**\*\*画图，学用circle画圆形。

**\*\*程序分析：**\*\*无。

```
1 from tkinter import *
2 canvas=Canvas(width=800,height=600,bg='yellow')
3 canvas.pack(expand=YES,fill=BOTH)
4 k=1
5 j=1
6 for i in range(26):
```

```
7     canvas.create_oval(310-k,250-k,310+k,250+k,width=1)
8     k+=j
9     j+=0.3
10 mainloop()
```

### 实例057：画线

**\*\*题目：**\*\*画图，学用line画直线。

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      from tkinter import *
3
4      canvas = Canvas(width=300, height=300, bg='green')
5      canvas.pack(expand=YES, fill=BOTH)
6      x0 = 263
7      y0 = 263
8      y1 = 275
9      x1 = 275
10     for i in range(19):
11         canvas.create_line(x0,y0,x0,y1, width=1, fill='red')
12         x0 = x0 - 5
13         y0 = y0 - 5
14         x1 = x1 + 5
15         y1 = y1 + 5
16
17     x0 = 263
18     y1 = 275
19     y0 = 263
20     for i in range(21):
21         canvas.create_line(x0,y0,x0,y1,fill = 'red')
22         x0 += 5
23         y0 += 5
24         y1 += 5
25
26     mainloop()
```

### 实例058：画矩形

**\*\*题目：**\*\*画图，学用rectangle画方形。

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      from tkinter import *
3      root = Tk()
4      root.title('Canvas')
5      canvas = Canvas(root,width = 400,height = 400,bg = 'yellow')
6      x0 = 263
7      y0 = 263
```

```

8     y1 = 275
9     x1 = 275
10    for i in range(19):
11        canvas.create_rectangle(x0,y0,x1,y1)
12        x0 -= 5
13        y0 -= 5
14        x1 += 5
15        y1 += 5
16
17    canvas.pack()
18    root.mainloop()

```

### 实例059：画图（丑）

**\*\*题目：**\*\*画图，综合例子。

**\*\*程序分析：**\*\*丑。

```

1  if __name__ == '__main__':
2      from tkinter import *
3      canvas = Canvas(width = 300,height = 300,bg = 'green')
4      canvas.pack(expand = YES,fill = BOTH)
5      x0 = 150
6      y0 = 100
7      canvas.create_oval(x0 - 10,y0 - 10,x0 + 10,y0 + 10)
8      canvas.create_oval(x0 - 20,y0 - 20,x0 + 20,y0 + 20)
9      canvas.create_oval(x0 - 50,y0 - 50,x0 + 50,y0 + 50)
10     import math
11     B = 0.809
12     for i in range(16):
13         a = 2 * math.pi / 16 * i
14         x = math.ceil(x0 + 48 * math.cos(a))
15         y = math.ceil(y0 + 48 * math.sin(a) * B)
16         canvas.create_line(x0,y0,x,y,fill = 'red')
17     canvas.create_oval(x0 - 60,y0 - 60,x0 + 60,y0 + 60)
18
19
20     for k in range(501):
21         for i in range(17):
22             a = (2 * math.pi / 16) * i + (2 * math.pi / 180) * k
23             x = math.ceil(x0 + 48 * math.cos(a))
24             y = math.ceil(y0 + 48 * math.sin(a) * B)
25             canvas.create_line(x0,y0,x,y,fill = 'red')
26         for j in range(51):
27             a = (2 * math.pi / 16) * i + (2 * math.pi / 180) * k - 1
28             x = math.ceil(x0 + 48 * math.cos(a))
29             y = math.ceil(y0 + 48 * math.sin(a) * B)
30             canvas.create_line(x0,y0,x,y,fill = 'red')
31     mainloop()

```

## 实例060：字符串长度

**\*\*题目：**\*\*计算字符串长度。

**\*\*程序分析：**\*\*无。

```
1 s='zhangguang101'
2 print(len(s))
```

## 实例061：杨辉三角

**\*\*题目：**\*\*打印出杨辉三角形前十行。

**\*\*程序分析：**\*\*无。

```
1 def generate(numRows):
2     r = [[1]]
3     for i in range(1,numRows):
4         r.append(list(map(lambda x,y:x+y, [0]+r[-1],r[-1]+[0])))
5     return r[:numRows]
6 a=generate(10)
7 for i in a:
8     print(i)
```

## 实例062：查找字符串

**\*\*题目：**\*\*查找字符串。

**\*\*程序分析：**\*\*无。

```
1 s1='aabbxuebixuebi'
2 s2='ab'
3 s3='xue'
4 print(s1.find(s2))
5 print(s1.find(s3))
```

## 实例063：画椭圆

**\*\*题目：**\*\*画椭圆。

**\*\*程序分析：**\*\*使用 tkinter。

```
1 if __name__ == '__main__':
2     from tkinter import *
3     x = 360
4     y = 160
5     top = y - 30
6     bottom = y - 30
7
8     canvas = Canvas(width = 400,height = 600,bg = 'white')
9     for i in range(20):
10         canvas.create_oval(250 - top,250 - bottom,250 + top,250 + bottom)
11         top -= 5
```

```
12         bottom += 5
13     canvas.pack()
14     mainloop()
```

### 实例64：画椭圆、矩形

**\*\*题目：**\*\*利用ellipse 和 rectangle 画图。。

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      from tkinter import *
3      canvas = Canvas(width = 400,height = 600,bg = 'white')
4      left = 20
5      right = 50
6      top = 50
7      num = 15
8      for i in range(num):
9          canvas.create_oval(250 - right,250 - left,250 + right,250 + left)
10         canvas.create_oval(250 - 20,250 - top,250 + 20,250 + top)
11         canvas.create_rectangle(20 - 2 * i,20 - 2 * i,10 * (i + 2),10 * (
12             i + 2))
13         right += 5
14         left += 5
15         top += 10
16
17     canvas.pack()
18     mainloop()
```

### 实例065：画组合图形

**\*\*题目：**\*\*一个最优美的图案。

**\*\*程序分析：**\*\*无。

```
1  import math
2  from tkinter import *
3
4  class PTS:
5      def __init__(self):
6          self.x = 0
7          self.y = 0
8  points = []
9
10 def LineToDemo():
11     screenx = 400
12     screeny = 400
13     canvas = Canvas(width = screenx,height = screeny,bg = 'white')
14
15     AspectRatio = 0.85
16     MAXPTS = 15
```

```

17     h = screeny
18     w = screenx
19     xcenter = w / 2
20     ycenter = h / 2
21     radius = (h - 30) / (AspectRatio * 2) - 20
22     step = 360 / MAXPTS
23     angle = 0.0
24     for i in range(MAXPTS):
25         rads = angle * math.pi / 180.0
26         p = PTS()
27         p.x = xcenter + int(math.cos(rads) * radius)
28         p.y = ycenter - int(math.sin(rads) * radius * AspectRatio)
29         angle += step
30         points.append(p)
31     canvas.create_oval(xcenter - radius,ycenter - radius,
32                       xcenter + radius,ycenter + radius)
33     for i in range(MAXPTS):
34         for j in range(i,MAXPTS):
35
36         canvas.create_line(points[i].x,points[i].y,points[j].x,points[j].y)
37
38     canvas.pack()
39     mainloop()
40 if __name__ == '__main__':
41     LineToDemo()

```

## 实例066：三数排序

**\*\*题目：**\*\*输入3个数a,b,c，按大小顺序输出。

**\*\*程序分析：**\*\*同实例005。

```

1 raw=[]
2 for i in range(3):
3     x=int(input('int%d:'%(i)))
4     raw.append(x)
5
6 for i in range(len(raw)):
7     for j in range(i,len(raw)):
8         if raw[i]>raw[j]:
9             raw[i],raw[j]=raw[j],raw[i]
10 print(raw)
11
12
13 raw2=[]
14 for i in range(3):
15     x=int(input('int%d:'%(i)))
16     raw2.append(x)
17 print(sorted(raw2))

```



### 实例067：交换位置

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

**\*\*程序分析：**\*\*无。

```
1  li=[3,2,5,7,8,1,5]
2
3  li[-1],li[li.index(min(li))]=li[li.index(min(li))],li[-1]
4
5  m=li[0]
6  ind=li.index(max(li))
7  li[0]=li[ind]
8  li[ind]=m
9
10 print(li)
```

### 实例068：旋转数列

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

**\*\*程序分析：**\*\*无。

```
1  from collections import *
2  li=[1,2,3,4,5,6,7,8,9]
3  deq=deque(li,maxlen=len(li))
4  print(li)
5  deq.rotate(int(input('rotate:')))
6  print(list(deq))
```

### 实例069：报数

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

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      nmax = 50
3      n = int(input('请输入总人数:'))
4      num = []
5      for i in range(n):
6          num.append(i + 1)
7
8      i = 0
9      k = 0
10     m = 0
11
12     while m < n - 1:
13         if num[i] != 0 : k += 1
```

```

14         if k == 3:
15             num[i] = 0
16             k = 0
17             m += 1
18             i += 1
19             if i == n : i = 0
20
21     i = 0
22     while num[i] == 0: i += 1
23     print(num[i])

```

### 实例070：字符串长度II

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

**\*\*程序分析：**\*\*无。

```

1 def lenofstr(s):
2     return len(s)
3
4 print(lenofstr('tanxiaofengsheng'))

```

### 实例071：输入和输出

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

**\*\*程序分析：**\*\*无。

```

1 N = 3
2 #stu
3 # num : string
4 # name : string
5 # score[4]: list
6 student = []
7 for i in range(5):
8     student.append(['', '', []])
9
10 def input_stu(stu):
11     for i in range(N):
12         stu[i][0] = input('input student num:\n')
13         stu[i][1] = input('input student name:\n')
14         for j in range(3):
15             stu[i][2].append(int(input('score:\n')))
16
17 def output_stu(stu):
18     for i in range(N):
19         print ('%-6s%-10s' % ( stu[i][0],stu[i][1] ))
20         for j in range(3):
21             print ('%-8d' % stu[i][2][j])
22

```

```
23 if __name__ == '__main__':
24     input_stu(student)
25     print (student)
26     output_stu(student)
```

实例072：创建链表

**\*\*题目：\*\***创建一个链表。

**\*\*程序分析：\*\***原文不太靠谱。

```
1  class Node:
2
3      def __init__(self, data):
4          self.data = data
5          self.next = None
6
7      def get_data(self):
8          return self.data
9
10 class List:
11
12     def __init__(self, head):
13         self.head = head
14
15     def is_empty(self):
16         return self.get_len() == 0
17
18     def get_len(self):
19         length = 0
20         temp = self.head
21         while temp is not None:
22             length += 1
23             temp = temp.next
24         return length
25
26     def append(self, node):
27         temp = self.head
28         while temp.next is not None:
29             temp = temp.next
30         temp.next = node
31
32     def delete(self, index):
33         if index < 1 or index > self.get_len():
34             print("给定位置不合理")
35             return
36         if index == 1:
37             self.head = self.head.next
38             return
39         temp = self.head
40         cur_pos = 0
```

```

41     while temp is not None:
42         cur_pos += 1
43         if cur_pos == index-1:
44             temp.next = temp.next.next
45             temp = temp.next
46
47     def insert(self, pos, node):
48         if pos < 1 or pos > self.get_len():
49             print("插入结点位置不合理")
50             return
51         temp = self.head
52         cur_pos = 0
53         while temp is not None:
54             cur_pos += 1
55             if cur_pos == pos-1:
56                 node.next = temp.next
57                 temp.next = node
58                 break
59             temp = temp.next
60
61     def reverse(self, head):
62         if head is None and head.next is None:
63             return head
64         pre = head
65         cur = head.next
66         while cur is not None:
67             temp = cur.next
68             cur.next = pre
69             pre = cur
70             cur = temp
71         head.next = None
72         return pre
73
74     def print_list(self, head):
75         init_data = []
76         while head is not None:
77             init_data.append(head.get_data())
78             head = head.next
79         return init_data
80
81 if __name__ == '__main__':
82     head = Node('head')
83     link = List(head)
84     for i in range(10):
85         node = Node(i)
86         link.append(node)
87     print(link.print_list(head))

```

### 实例073：反向输出链表

**\*\*题目：**\*\*反向输出一个链表。

**\*\*程序分析：**\*\*无。

```
1 class Node:
2
3     def __init__(self, data):
4         self.data = data
5         self.next = None
6
7     def get_data(self):
8         return self.data
9
10 class List:
11
12     def __init__(self, head):
13         self.head = head
14
15     def is_empty(self):
16         return self.get_len() == 0
17
18     def get_len(self):
19         length = 0
20         temp = self.head
21         while temp is not None:
22             length += 1
23             temp = temp.next
24         return length
25
26     def append(self, node):
27         temp = self.head
28         while temp.next is not None:
29             temp = temp.next
30         temp.next = node
31
32     def delete(self, index):
33         if index < 1 or index > self.get_len():
34             print("给定位置不合理")
35             return
36         if index == 1:
37             self.head = self.head.next
38             return
39         temp = self.head
40         cur_pos = 0
41         while temp is not None:
42             cur_pos += 1
43             if cur_pos == index-1:
44                 temp.next = temp.next.next
45                 temp = temp.next
46
```

```

47     def insert(self, pos, node):
48         if pos < 1 or pos > self.get_len():
49             print("插入结点位置不合理")
50             return
51         temp = self.head
52         cur_pos = 0
53         while temp is not None:
54             cur_pos += 1
55             if cur_pos == pos-1:
56                 node.next = temp.next
57                 temp.next = node
58                 break
59             temp = temp.next
60
61     def reverse(self, head):
62         if head is None and head.next is None:
63             return head
64         pre = head
65         cur = head.next
66         while cur is not None:
67             temp = cur.next
68             cur.next = pre
69             pre = cur
70             cur = temp
71         head.next = None
72         return pre
73
74     def print_list(self, head):
75         init_data = []
76         while head is not None:
77             init_data.append(head.get_data())
78             head = head.next
79         return init_data
80
81 if __name__ == '__main__':
82     head = Node('head')
83     link = List(head)
84     for i in range(10):
85         node = Node(i)
86         link.append(node)
87     print(link.print_list(head))
88     print(link.print_list(link.reverse(head)))

```

### 实例074：列表排序、连接

**\*\*题目：**\*\*列表排序及连接。

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

```

1 a=[2,6,8]
2 b=[7,0,4]
3 a.extend(b)
4 a.sort()
5 print(a)

```

实例075：不知所云

**\*\*题目：**\*\*放松一下，算一道简单的题目。

**\*\*程序分析：**\*\*鬼知道是什么。

```

1 if __name__ == '__main__':
2     for i in range(5):
3         n = 0
4         if i != 1: n += 1
5         if i == 3: n += 1
6         if i == 4: n += 1
7         if i != 4: n += 1
8         if n == 3: print (64 + i)

```

## 实例076：做函数

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

**\*\*程序分析：**\*\*无。

```

1 def peven(n):
2     i = 0
3     s = 0.0
4     for i in range(2,n + 1,2):
5         s += 1.0 / i
6     return s
7
8 def podd(n):
9     s = 0.0
10    for i in range(1, n + 1,2):
11        s += 1.0 / i
12    return s
13
14 def dcall(fp,n):
15     s = fp(n)
16     return s
17
18 if __name__ == '__main__':
19     n = int(input('input a number: '))
20     if n % 2 == 0:
21         sum = dcall(peven,n)
22     else:
23         sum = dcall(podd,n)
24     print (sum)

```

## 实例077：遍历列表

**\*\*题目：**\*\*循环输出列表

**\*\*程序分析：**\*\*无。

```
1 l=['moyu','niupi','xuecaibichi','shengfaji','42']
2 for i in range(len(l)):
3     print(l[i])
```

## 实例078：字典

**\*\*题目：**\*\*找到年龄最大的人，并输出。请找出程序中有什么问题。

**\*\*程序分析：**\*\*无。

```
1 if __name__ == '__main__':
2     person = {"li":18,"wang":50,"zhang":20,"sun":22}
3     m = 'li'
4     for key in person.keys():
5         if person[m] < person[key]:
6             m = key
7
8     print ('%s,%d' % (m,person[m]))
```

## 实例079：字符串排序

**\*\*题目：**\*\*字符串排序。

**\*\*程序分析：**\*\*无。

```
1 l=['baaa','aaab','aaba','aaaa','abaa']
2 l.sort()
3 print(l)
```

## 实例080：猴子分桃

**\*\*题目：**\*\*海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子平均分为五份，多了一个，这只猴子把多的一个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，又多了个，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。

```
1 if __name__ == '__main__':
2     i = 0
3     j = 1
4     x = 0
5     while (i < 5) :
6         x = 4 * j
7         for i in range(0,5) :
```



```

8         if(x%4 != 0) :
9             break
10        else :
11            i += 1
12            x = (x/4) * 5 +1
13        j += 1
14    print(x)
15
16    for p in range(5):
17        x=(x-1)/5*4
18    print(x)

```

### 实例081：求未知数

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

\*\*程序分析：\*\*无。

```

1  a = 809
2  for i in range(10,100):
3      b = i * a
4      if b >= 1000 and b <= 10000 and 8 * i < 100 and 9 * i >= 100:
5          print(b,' = 800 * ', i, ' + 9 * ', i)
6
7
8  for i in range(10,100):
9      if 8*i>99 or 9*i<100:
10         continue
11     if 809*i==800*i+9*i:
12         print(i)
13         break

```

### 实例082：八进制转十进制

\*\*题目：\*\*八进制转换为十进制

\*\*程序分析：\*\*无。

```

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

```

### 实例083：制作奇数

\*\*题目：\*\*求0—7所能组成的奇数个数。

程序分析：

组成1位数是4个。1,3,5,7结尾

组成2位数是7\*4个。第一位不能为0

组成3位数是7\*4\*4个。中间随意

组成4位数是788\*4个。

```
1  if __name__ == '__main__':
2      sum = 4
3      s = 4
4      for j in range(2,9):
5          print (sum)
6          if j <= 2:
7              s *= 7
8          else:
9              s *= 8
10         sum += s
11     print('sum = %d' % sum)
```

#### 实例084：连接字符串

**\*\*题目：**\*\*连接字符串。

**\*\*程序分析：**\*\*无。

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

#### 实例085：整除

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

**\*\*程序分析：**\*\*999999 / 13 = 76923。

```
1  if __name__ == '__main__':
2      zi = int(input('输入一个数字:'))
3      n1 = 1
4      c9 = 1
5      m9 = 9
6      sum = 9
7      while n1 != 0:
8          if sum % zi == 0:
9              n1 = 0
10         else:
11             m9 *= 10
12             sum += m9
13             c9 += 1
14     print ('%d 个 9 可以被 %d 整除 : %d' % (c9,zi,sum))
15     r = sum / zi
16     print ('%d / %d = %d' % (sum,zi,r))
```

#### 实例086：连接字符串II

**\*\*题目：**\*\*两个字符串连接程序。

**\*\*程序分析：**\*\*无。

```

1 a='guangtou'
2 b='feipang'
3 print(b+a)

```

#### 实例087：访问类成员

**\*\*题目：**\*\*回答结果（结构体变量传递）。

**\*\*程序分析：**\*\*无。

```

1 if __name__ == '__main__':
2     class student:
3         x = 0
4         c = 0
5     def f(stu):
6         stu.x = 20
7         stu.c = 'c'
8     a= student()
9     a.x = 3
10    a.c = 'a'
11    f(a)
12    print(a.x,a.c)

```

#### 实例088：打印星号

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

**\*\*程序分析：**\*\*无。

```

1 for i in range(3):
2     print('*'*int(input('input a number: ')))

```

#### 实例089：解码

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

**\*\*程序分析：**\*\*无。

```

1 n=input()
2 n = str(n)
3 a=[]
4 for i in range(4):
5     a.append((int(n[i])+5))
6 a[0],a[3]=a[3],a[0]
7 a[1],a[2]=a[2],a[1]
8 print ("".join('%s' %s for s in a))
9

```

## 实例090：列表详解

**\*\*题目：**\*\*列表使用实例。

**\*\*程序分析：**\*\*无。

```
1  #list
2  #新建列表
3  testList=[10086,'中国移动',[1,2,4,5]]
4
5  #访问列表长度
6  print (len(testList) )
7  #到列表结尾
8  print (testList[1:])
9  #向列表添加元素
10 testList.append('i\'m new here!')
11
12 print (len(testList) )
13 print (testList[-1] )
14 #弹出列表的最后一个元素
15 print (testList.pop(1) )
16 print (len(testList) )
17 print (testList )
18 #list comprehension
19 #后面有介绍，暂时掠过
20 matrix = [[1, 2, 3],
21            [4, 5, 6],
22            [7, 8, 9]]
23 print (matrix )
24 print (matrix[1] )
25 col2 = [row[1] for row in matrix]#get a column from a matrix
26 print (col2 )
27 col2even = [row[1] for row in matrix if row[1] % 2 == 0]#filter odd item
28 print (col2even)
```

## 实例091：time模块

**\*\*题目：**\*\*时间函数举例1。

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      import time
3      print (time.ctime(time.time()))
4      print (time.asctime(time.localtime(time.time())))
5      print (time.asctime(time.gmtime(time.time())))
```

## 实例092：time模块II

**\*\*题目：**\*\*时间函数举例2。

**\*\*程序分析：**\*\*如何浪费时间。

```
1  if __name__ == '__main__':
2      import time
3      start = time.time()
4      for i in range(3000):
5          print(i)
6      end = time.time()
7
8      print (end - start)
```

### 实例093: time模块III

**\*\*题目：**\*\*时间函数举例3。

**\*\*程序分析：**\*\*如何浪费时间。

```
1  if __name__ == '__main__':
2      import time
3      start = time.clock()
4      for i in range(100):
5          print(i)
6      end = time.clock()
7      print('different is %6.3f' % (end - start))
```

### 实例094: time模块IV

**\*\*题目：**\*\*时间函数举例4。

**\*\*程序分析：**\*\*如何浪费时间。

```
1  if __name__ == '__main__':
2      import time
3      import random
4
5      play_it = input('do you want to play it.(\ny\' or \'n\')')
6      while play_it == 'y':
7          c = input('input a character:\n')
8          i = random.randint(0,2**32) % 100
9          print ('please input number you guess:\n')
10         start = time.clock()
11         a = time.time()
12         guess = int(input('input your guess:\n'))
13         while guess != i:
14             if guess > i:
15                 print('please input a little smaller')
16                 guess = int(input('input your guess:\n'))
17             else:
18                 print('please input a little bigger')
19                 guess = int(input('input your guess:\n'))
20         end = time.clock()
21         b = time.time()
```

```

22     var = (end - start) / 18.2
23     print (var)
24     # print 'It took you %6.3 seconds' % time.diffftime(b,a))
25     if var < 15:
26         print ('you are very clever!')
27     elif var < 25:
28         print ('you are normal!')
29     else:
30         print ('you are stupid!')
31     print ('Congradulations')
32     print ('The number you guess is %d' % i)
33     play_it = input('do you want to play it.')

```

### 实例095：转换时间格式

**\*\*题目：**\*\*字符串日期转换为易读的日期格式。

**\*\*程序分析：**\*\*看看就得了，dateutil是个第三方库。

```

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

```

### 实例096：计算复读次数

**\*\*题目：**\*\*计算字符串中子串出现的次数。

**\*\*程序分析：**\*\*无。

```

1  s1='xuebixuebixuebixuebixuebixuebixue'
2  s2='xuebi'
3  print(s1.count(s2))

```

### 实例097：磁盘写入

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

**\*\*程序分析：**\*\*无。

```

1  if __name__ == '__main__':
2      from sys import stdout
3      filename = input('输入文件名:\n')
4      fp = open(filename,"w")
5      ch = input('输入字符串:\n')
6      while ch != '#':
7          fp.write(ch)
8          stdout.write(ch)
9          ch = input('')
10     fp.close()

```

## 实例098：磁盘写入II

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

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      fp = open('test.txt','w')
3      string = input('please input a string:\n')
4      string = string.upper()
5      fp.write(string)
6      fp = open('test.txt','r')
7      print (fp.read())
8      fp.close()
```

## 实例099：磁盘读写

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

**\*\*程序分析：**\*\*无。

```
1  if __name__ == '__main__':
2      import string
3      fp = open('test1.txt')
4      a = fp.read()
5      fp.close()
6
7      fp = open('test2.txt')
8      b = fp.read()
9      fp.close()
10
11     fp = open('test3.txt','w')
12     l = list(a + b)
13     l.sort()
14     s = ''
15     s = s.join(l)
16     fp.write(s)
17     fp.close()
```

## 实例100：列表转字典

**\*\*题目：**\*\*列表转换为字典。

**\*\*程序分析：**\*\*无。

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