

# Python题库

## 1、求最大公约数和最小公倍数

两个数的最大公约数是两个数的公共因子中最大的那个数；两个数的最小公倍数则是能够同时被两个数整除的最小的那个数。

输入：（120和60是通过键盘输入的）

x = 120

y = 60

输出：

60和120的最大公约数是：60

60和120的最小公倍数是：120

答案：

```
1 x = int(input('x = '))
2 y = int(input('y = '))
3 # 如果x大于y就交换x和y的值
4 if x > y:
5     # 通过下面的操作将y的值赋给x，将x的值赋给y
6     temp = y
7     y = x
8     x = temp
9     # 从两个数中较的数开始做递减的循环
10 for factor in range(x, 0, -1):
11     if x % factor == 0 and y % factor == 0:
12         print('{}和{}的最大公约数是：{}'.format(x, y, factor))
13         print('{}和{}的最小公倍数是：{}'.format(x, y, x * y // factor))
14         break
```

关卡：1、2、3、4、5、6

难度：容易

## 2、斐波那契数列

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：

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：

$F_0 = 0$  (n=0)

$F_1 = 1 \quad (n=1)$

$F_n = F_{n-1} + F_{n-2} \quad (n \geq 2)$

输入：（数字10是通过键盘输入的）

请输入需要打印的斐波那契数列的数据个数（从1开始）：10

输出：（每一个数据项之间用空格）

1 1 2 3 5 8 13 21 34 55

答案：

```
1 def fib(n):
2     a, b = 1, 1
3     #通过循环来打印数据
4     for i in range(n):
5         print(a, end = " ")
6         temp = a
7         a = b
8         b = temp+b
9     num = int(input("请输入需要打印的斐波那契数列的数据个数（从1开始）："))
10    fib(num)
```

关卡：1、2、3、4、5、6、7

难度：容易

### 3、求和运算

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

输入：

$a = 4$

$n = 5$

输出：

4

44

444

4444

44444

前5个数的和为：49380

答案：

```
1 Tn = 0
2 Sn = []
3 a = int(input('a = '))
4 n = int(input('n = '))
5 for count in range(n):
```

```

6     Tn = Tn + a
7     a = a * 10
8     Sn.append(Tn)
9     print(Tn)
10 print("前{}个数的和为: {}".format(n, sum(Sn)))

```

关卡：1、2、3、4、5、6、7、8

难度：容易

#### 4、求前n阶乘的和

比如求 $1+2!+3!+\dots+20!$ 的和。

输入：（不要输入太大的n防止数据太大，电脑卡死）

n = 20

输出：

前20阶乘的和是：2561327494111820313

答案：

```

1 n = int(input("n = "))
2 s = 0
3 t = 1
4 for i in range(1,n+1):
5     t *= i
6     s += t
7 print ("前{}阶乘的和是: {}".format(n,s))

```

关卡：1、2、3、4、5、6、7

难度：容易

#### 5、求年龄

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？

输入：无

输出：第五个人的年龄

答案：

```

1 def age(n):
2     if n == 1:
3         c = 10
4     else:
5         c = age(n - 1) + 2
6     return c
7 print ("第五个人的年龄是: ",age(5))

```

关卡：1、2、3、4、5、6、7、20

难度：容易。

## 6、求两个列表中的相同元素和不同元素

给定两个列表，怎么找出他们相同的元素和不同的元素？

输入：

list1 = [1,2,3]

list2 = [3,4,5]

输出：

list1和list2的相同元素是： {3}

list1和list2的不同元素是： {1, 2, 4, 5}

答案：

```
1 list1 = [1,2,3]
2 list2 = [3,4,5]
3 set1 = set(list1)
4 set2 = set(list2)
5 print("list1和list2的相同元素是: ",set1 & set2)
6 print("list1和list2的不同元素是: ",set1 ^ set2)
```

关卡：1、2、3、8

难度：容易

## 7、列表转换为字典

两个列表转合并为一个字典

输入：

m = ['a', 11]

n = ['b', 22]

输出：

{'a':11,'b':22}

答案：

```
1 m = ['a', 11]
2 n = ['b', 22]
3 print(dict([m,n]))
```

关卡：1、2、3、8

难度：容易

## 8、倒序输出列表中的数据

按相反的顺序输出列表的值。

输入：

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

输出：

three

two

one

答案:

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

关卡: 1、2、3、8

难度: 容易

## 9、百钱百鸡问题

百钱百鸡是我国古代数学家张丘建在《算经》一书中提出的数学问题: 鸡翁一值钱五, 鸡母一值钱三, 鸡雏三值钱一。百钱买百鸡, 问鸡翁、鸡母、鸡雏各几何? 翻译成现代文是: 公鸡5元一只, 母鸡3元一只, 小鸡1元三只, 用100块钱买一百只鸡, 问公鸡、母鸡、小鸡各有多少只?

输入:

无

输出: (有多个答案, 下面的只是其中一个答案)

公鸡: 0只, 母鸡: 25只, 小鸡: 75只

答案:

```
1 for x in range(0, 20):
2     for y in range(0, 33):
3         z = 100 - x - y
4         if 5 * x + 3 * y + z / 3 == 100:
5             print('公鸡: {}只, 母鸡: {}只, 小鸡: {}只'.format (x, y, z))
```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 10、翻转数字

通过键盘输入一个数字, 若

(1) 该数字各位上的数字之和为奇数, 则将该数各位数倒叙打印 (如122 (各个位上的数字之和为 $1+2+2=5$ ), 打印221)

(2) 若该数字各位数之和为偶数, 则直接打印该数字  
注意这里的整数可以是负数。请参考下面的两个输出。

输入与输出:

终端

```
请输入一个整数:221
num2=221
122
□
```

终端

```
请输入一个整数:-221
num2=221
-122
□
```

答案:

```
1 num1 = input('请输入一个整数:')
2 num3 = int(num1)
3 if num3>0:
4     len_num = len(num1)
5 else:
6     len_num = len(num1)-1
7 if num3<0:
8     num2=-num3
9 else:
10    num2=num3
11 i,sum=1,0
12 num2_str = ""
13 print("num2="+str(num2))
14 while i<=len_num:
15     sum+= num2%10
16     num2_str += str(num2%10)
17     num2 = num2 // (10)
18     i+=1
19 if sum%2 ==0:
20     print(num1)
21 else:
22     if(num3>=0):
23         print(num2_str)
24     else:
25         print("-" + num2_str)
```

关卡: 1、2、3、4、5、6、7、8

难度: 中等

## 11、身体BMI质量检测

身体质量指数（BMI），是目前国际上常用的衡量人体胖瘦程度以及是否健康的一个指标。它的计算公式： $BMI = \text{体重} \div \text{身高}^2$ （注意是体重除以身高的平方）。其中，体重的单位是kg，身高的单位是m。中国人的BMI参考标准：BMI<18.5为偏瘦； $18.5 \leq BMI < 24$ 为正常； $24 \leq BMI < 28$ 为偏胖； $BMI \geq 28$ 为肥胖。编写一个程序，输入一个人的体重和身高，计算BMI并输出体重是否正常。请在下面回答处上传程序的代码和运行结果。（可以在课程中任选一个代码框编写代码），如下图是一个运行结果的示例，其中74和1.74都是通过键盘输入的。（本题18分，涉及的知识包括：if-elif结构，float()函数，input()函数，比较运算符，字符串格式化输出等）

输入与输出如下：

```
请输入体重 (kg): 74
请输入身高 (m): 1.74
你的bmi是: 24.44180208746202
超重
□
```

答案：

```
1 mass = float(input('请输入体重(kg): '))
2 height = float(input('请输入身高(m): '))
3 bmi = mass / height / height
5 if bmi < 18.5 :
6     body = '偏瘦'
7 elif 18.5 <= bmi and bmi < 24 :
8     body = '正常'
9 elif 24 <= bmi and bmi < 28 :
10    body = '超重'
11 elif bmi >= 28 :
12    body = '肥胖'
13 print('你的bmi是: {}'.format(bmi))
14 print(body)
```

关卡：1、2、3、4、5、6、7

难度：中等

## 12、实例和实例方法的创建

使用Python的class定义一个类，它具有两个int的属性，a和b，有一个sum方法，得到a和b的和。初始化这个类的一个示例，并打印出它的两个属性和sum方法。

答案：

```
1 class Basic:
2     def __init__(self, a, b):
3         self.a = a
4         self.b = b
```

```

5     def sum(self):
6         return self.a + self.b
7     i = Basic(1, 2)
8     print (i.a, i.b, i.sum())

```

关卡：1、2、3、4、5、9、10、14、

难度：中等

### 13、100以内的偶数和

用for循环实现1~100之间的偶数求和。

输入：无

输出：1到100之间的偶数和是： 2550

答案：

```

1 sum = 0
2 for x in range(1, 101):
3     if x % 2 == 0:
4         sum += x
5 print('1到100之间的偶数和是：',sum)

```

关卡：1、2、3、4、5、6、7

难度：容易

### 14、输入年份判断是不是闰年

通过键盘输入年份，如果是闰年输出True 否则输出False

输入：

请输入年份: 1900

输出：

1900年不是闰年

答案：

```

1 year = int(input('请输入年份：'))
2 if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
3     print("{}年是闰年".format(year))
4 else:
5     print("{}年不是闰年".format(year))

```

关卡：1、2、3、4、5、6、7

难度：容易

### 15、分段函数

编写程序，实现分段函数的计算如下：

x	y
x<0	0



$0 \leq x < 5$	$x$
$5 \leq x < 10$	$3x - 5$
$10 \leq x < 20$	$0.5x - 2$
$20 \leq x$	$0$

输入：

通过键盘输入x的值。

$x = 10$

输出：

根据上表计算y的值

$y = 3$

答案：

```

1 x = eval(input("x = "))
2 if x < 0:
3     y = 0
4 elif 0 <= x < 5:
5     y = x
6 elif 5 <= x < 10:
7     y = 3 * x - 5
8 elif 10 <= x < 20:
9     y = 0.5 * x - 2
10 elif 20 <= x:
11     y = 0
12 print("y = {}".format(y))

```

关卡：1、2、3、4、5、6、7

难度：简单

## 16、因式分解

用户从键盘输入小于1000的整数，对齐进行因式分解，比如 $10=2*5$ ， $60 = 2*2*3*5$

输入：

请输入一个小于1000整数：90

输出：

$90 = 2*3*3*5$

答案：

```

1 x = input("请输入一个小于1000的整数：")
2 x = eval(x)
3 t = x
4 i = 2
5 result = []
6 while True:
7     if t == 1:
8         break

```

```

9         if t % i == 0:
10             result.append(i)
11             t = t / i
12         else:
13             i += 1
14     print(x, "=", " ".join(map(str, result)))

```

关卡：1、2、3、4、5、6、7、8

难度：中等

## 17、列表排序

生成一个包含20个随机整数（100以内的整数）的列表，对其中偶数索引（下标）的数据进行降序排列，奇数索引（下标）的元素不变。

输入：无

输出：（下面是一种输出情况）

排序以前的结果是：[94, 15, 52, 20, 45, 75, 80, 78, 18, 68, 32, 75, 35, 55, 38, 71, 62, 40, 16, 56]

排序以后的结果是：[94, 15, 80, 20, 62, 75, 52, 78, 45, 68, 38, 75, 35, 55, 32, 71, 18, 40, 16, 56]

答案：

```

1 import random
2 x = [random.randint(0,100) for i in range(20)]
3 print("排序以前的结果是：{}".format(x))
4 x[::2] = sorted(x[::2],reverse=True)
5 print("排序以后的结果是：{}".format(x))

```

关卡：1、2、3、4、5、6、7、8、13

难度：困难

## 18、删除列表中的奇数

生成一个包含20个随机整数（100以内的整数）的列表，然后删除其中的所有奇数。

输入：

无

输出：

删除奇数以前的结果是：[58, 34, 84, 42, 28, 10, 80, 72, 91, 27, 51, 6, 21, 13, 50, 34, 81, 5, 93, 100]

删除奇数以后的结果是：[58, 34, 84, 42, 28, 10, 80, 72, 6, 50, 34, 100]

答案：

```

1 import random
2 x = [random.randint(0, 100) for i in range(20)]
3 print("删除奇数以前的结果是：{}".format(x))
4 for i in range(len(x))[::-1]:

```

```
5     if x[i] % 2 == 1:
6         x.remove(x[i])
7     print("删除奇数以后的结果是: {}".format(x))
```

关卡：1、2、3、4、5、6、7、8

难度：中等

## 19、得到新列表

编写程序，输入一个包含若干自然数的列表，输出一个新列表，列表中的每个元素为原列表中每个自然数的位数，比如输入：[1,888,99,23456],输出：[1,3,2,5]

输入：

请输入一个包含若干自然数的列表：[1,888,99,23456]

输出：

[1,3,2,5]

答案：

```
1 data = eval(input("请输入一个包含若干自然数的列表："))
2 temp = []
3 for i in data:
4     temp.append(len(str(i)))
5 print(temp)
```

关卡：1、2、3、4、5、6、7、8

难度：中等

## 20、列表求均值

编写程序，输入一个包含若干自然数的列表，输出这些自然数的平均值，结果保留3位小数。

输入：

请输入一个包含若干自然数的列表：[2,3,4]

输出：

3.000

答案：

```
1 data = eval(input("请输入一个包含若干自然数的列表："))
2 print('{:.3f}'.format(sum(data)/len(data)))
```

关卡：1、2、3、4、5、6、7、8

难度：容易

## 21、求内积

编写程序，输入两个包含若干整数的等长列表，把这个两个列表看做两个向量，输出这两个向量的内积。，比如[1,2,3]和[4,5,6]的内积就是：1\*4+2\*5+3\*6的结果就是32

输入：

请输入一个包含若干整数的列表：[1,2,3]

请再输入一个包含若干整数的等列表: [4,5,6]

输出:

32

答案:

```
1 vector1 = eval(input("请输入一个包含若干整数的列表: "))
2 vector2 = eval(input("请再输入一个包含若干整数的等列表: "))
3 temp = 0
4 for i in range(len(vector1)):
5     temp = temp+vector1[i]*vector2[i]
6 print(temp)
```

关卡: 1、2、3、4、5、6、7、8

难度: 容易

## 22、求列表中等价于True的数据

编写程序, 输入一个包含任意数据的列表, 输出该列表中等价于True的元素组成的列表。例如: 输入: [1,2,3,0,', 'a',False], 输出: [1,2,3,'a']

输入:

请输入一个包含若干任意元素的列表: [1,2,3,0,', 'a',False]

输出:

[1, 2, 3, 'a']

答案:

```
1 data = eval(input("请输入一个包含若干任意元素的列表: "))
2 for i in range(len(data))[:-1]:
3     if bool(data[i]) == False:
4         data.remove(data[i])
5 print(data)
```

关卡: 1、2、3、4、5、6、7、8

难度: 容易

## 23、求列表中绝对值最大的数据

编写程序, 输入一个包含若干实数的列表, 输出其中绝对值最大的实数, 例如, 输入: [-8,64,3.5,-89], 输出: -89

输入:

请输入一个包含若干实数的列表: [-8,64,3.5,-89]

输出:

-89

答案:

```
1 data = eval(input("请输入一个包含若干实数的列表: "))
2 print(max(data,key = abs))
```

关卡: 1、2、3、4、5、6、7、8

难度: 中等

## 24、两个变量值互换

实现互换两个整数变量的值的功能。

输入：

x = 123

y = 456

输出：

值互换后x的值是：456

值互换后y的值是：123

答案：

```
1 x = int(input("x = "))
2 y = int(input("y = "))
3 x, y = y, x
4 print("值互换后x的值是：{}".format(x))
5 print("值互换后y的值是：{}".format(y))
```

关卡：1、2、3、4、5、6、7

难度：简单

## 25、最少多少个9能整除输入的奇数

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。比如，输入一整数13，能整除13的所有位都是数字都是9的整数是999999

输入：（不要输入偶数以免程序进入死循环卡死）

输入一个奇数:13

输出：

6 个 9 可以被 13 整除 : 999999

999999 / 13 = 76923

答案：

```
1 zi = int(input('输入一个奇数:'))
2 n1 = 1
3 c9 = 1
4 m9 = 9
5 sum = 9
6 while n1 != 0:
7     if sum % zi == 0:
8         n1 = 0
9     else:
10         m9 *= 10
11         sum += m9
12         c9 += 1
13 print('{} 个 9 可以被 {} 整除 : {}'.format(c9, zi, sum))
14 r = sum / zi
15 print('{} / {} = {}'.format(sum, zi, int(r)))
```

关卡：1、2、3、4、5、6、7、8、9

难度：困难

## 26、字符串转换为字典

将字符串 "k:1 |k1:2|k2:3|k3:4", 处理成字典 {k:1,k1:2,...}

输入: 无

输出:

{'k': 1, 'k1': 2, 'k2': 3, 'k3': 4}

答案:

```
1 str1 = "k:1|k1:2|k2:3|k3:4"
2 def str2dict(str1):
3     dict1 = {}
4     for iterm in str1.split('|'):
5         key,value = iterm.split(':')
6         dict1[key] = value
7     return dict1
8 #字典推导式
9 d = {k:int(v) for t in str1.split("|") for k, v in (t.split(":"), )}
10 print(d)
```

关卡: 1、2、3、4、5、6、7、8、9、10、11、12、13

难度: 困难

## 27、列表推导式的计算

生成一个列表, 列表中的数据是以整数1为第一个数据, 公差为n的等差数列, n是通过键盘输入的。

输入:

n = 20

输出:

[1, 21, 41, 61, 81, 101, 121, 141, 161, 181]

答案:

```
1 n = int(input("n = "))
2 print([1+x*n for x in range(10)])
```

关卡: 1、2、3、4、5、6、7、8

难度: 容易

## 28、验证码生成

设计一个函数产生指定长度的验证码, 每一位的验证码由大小写字母和数字构成, 不同位上的验证码可以重复, 比如验证码: qqwww。

输入:

验证码长度是: 4

输出: (因为是随机产生的验证码, 每个人的产生的结果可能都不一样)

4QMK

答案:

```
1 import random
2 def generate_code(code_len):
```

```

4     all_chars =
      '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
5     last_pos = len(all_chars) - 1
6     code = ''
7     for i in range(code_len):
8         index = random.randint(0, last_pos)
9         code += all_chars[index]
10    return code
11    n = int(input("验证码的长度是: "))
12    print(generate_code(n))

```

关卡：1、2、3、4、5、6、7、8、9、10、11、12、13

难度：中等

## 29、按要求输出1000以内的数

输出1000以内，对3、5、7整除，余数是2的所有正整数

输入：无

输出：

2  
107  
212  
317  
422  
527  
632  
737  
842  
947

答案：

```

1 for i in range(1, 1001):
2     if i % 3 == 2 and i % 5 == 2 and i % 7 == 2:
3         print(i)

```

关卡：1、2、3、4、5、6、7

难度：容易

## 30、不同类型的高校数量统计

列表ls中存储了先前的39所985高校所对应的学校类型，请以这个列表为数据变量，完善Python代码，统计输出各类型的高校数量。

```

1 ls = ["综合", "理工", "综合", "综合", "综合", "综合", "综合", "综合", "综合", "综合",
      "综合", "师范", "理工", "综合", "理工", "综合", "综合", "综合", "综合", "综
      合", "理工", "理工", "理工", "理工", "师范", "综合", "农林", "理工", "综合", "理
      工", "理工", "理工", "综合", "理工", "综合", "综合", "理工", "农林", "民族", "军
      事"]

```

输入：无

输出:

综合:20

理工:13

师范:2

农林:2

民族:1

军事:1

答案:

```
1 ls = ["综合", "理工", "综合", "综合", "综合", "综合", "综合", "综合", "综合",  
"综合", "师范", "理工", "综合", "理工", "综合", "综合", "综合", "综合", "综  
合", "理工", "理工", "理工", "理工", "师范", "综合", "农林", "理工", "综合", "理  
工", "理工", "理工", "综合", "理工", "综合", "综合", "理工", "农林", "民族", "军  
事"]  
2 d = {}  
3 for word in ls:  
4     d[word] = d.get(word, 0) + 1  
5 for k in d:  
6     print("{}:{}".format(k, d[k]))
```

关卡: 1、2、3、4、5、6、7、8

难度: 中等

### 31、谁是凶手

在日本某地发生了一起谋杀案, 警方通过排除确定杀人凶手必为四个嫌疑犯中的一个, 被控制的四个嫌疑犯的公司如下:

甲说: “不是我。”

乙说: “是丙。”

丙说: “是丁。”

丁说: “丙在胡说。”

已知三个人说了真话, 一个人说的是假话。现在请你根据这些信息, 找出到第谁是凶手。

可以用1、2、3、4分别表示甲、乙、丙、丁

输入:

无

输出:

杀手的编号是 3

True True False True

(杀手是丙, 丙在说谎)

答案:

```
1 '''  
2 用1~4分别表示4个嫌疑犯, 用枚举法列举杀手编号。  
3 判断如果4个人说的话有3句成立, 则找到答案。  
4 '''  
5 for k in range(1, 5):  
6     a = k != 1  
7     b = k == 3  
8     c = k == 4  
9     d = k != 4
```



```
10     if a + b + c + d == 3:
11         print('杀手的编号是', k)
12         print(a,b,c,d)
```

关卡：1、2、3、4、5、6、7、8

难度：容易

### 32、求整数的绝对值

不借助内部函数计算整数的绝对值。

输入：

请输入一个数:-12.4

输出：

12.4

答案：

```
1 x = float(input('请输入一个数:'))
2 if x < 0:
3     y = 0 - x
4 else:
5     y = x
6 print(y)
```

关卡：1、2、3、4、5、6、7

难度：容易

### 33、修改字典的value

有一个字典，保存的是学生各个编程语言的成绩，内容如下：

```
1 data = {
2     'python': 90,
3     'c++': 95,
4     'java': 90
5 }
```

修改'python'这个键对应的值为100，增加一个键值对，键是'php'，对应的值是92

输入：无

输出：

{'python': 100, 'c++': 95, 'java': 90, 'php': 92}

答案：

```
1 data = {
2     'python': 90,
3     'c++': 95,
4     'java': 90
5 }
6 data["python"] = 100
7 data["php"] = 92
8 print(data)
```

关卡：1、2、3、4、5、6、7

难度：简单

### 34、绘制五角星

学习完了最后一关turtle海龟绘图以后，利用turtle库，使用用循环结构绘制一个五角星

输入：无

输出：

一个五角星图案

答案：

```
1 from turtle import *
2 pensize(2)
3 seth(108)
4 for i in range(5):
5     left(144)
6     fd(100)
7     right(72)
8     fd(100)
```

关卡：1、2、3、4、5、6、7、20

难度：中等

### 35、成绩转换

把百分制的成绩转换为等级，规则是：

60分以下不合格；60~69分为合格；70~89分为良好；90分一行为优秀。当输入错误数据时，提示错误数据。

设计一个自动转换程序，输入一个百分制成绩，然后输出等级

输入：

请输入成绩：85

输出：

良好

答案：

```
1 score = int(input('请输入成绩: '))
2 if score < 60:
3     print('不合格')
4 elif 60 <= score <= 69:
5     print('合格')
6 elif 70 <= score <= 89:
7     print('良好')
8 elif 90 <= score <= 100:
9     print('优秀')
10 else:
11     print('错误数据')
```

关卡：1、2、3、4、5、6、7

难度：简单

### 36、达到珠穆朗玛峰的高度

假设有一张厚度0.5mm、面积足够大的纸。把这张纸不断对折，请问对折多少次后，可以达到珠穆朗玛峰的高度（8848m）。请编写循环结构的程序求解答案。

输入：无

输出：

25

答案：

```
1 h = 8848 * 1000
2 n = 0.5
3 t = 0
4 while True:
5     n = n * 2
6     t += 1
7     if n >= h:
8         break
9 print(t)
```

关卡：1、2、3、4、5、6、7、8、9

难度：容易

### 37、寻找和是10的数

从下面两个列表中各取一个数据，如果这俩数据的和是10，那么就把这俩数据组成一个元组显示出来。

```
1 lst1 = [3, 6, 1, 8, 1, 9, 2]
2 lst2 = [3, 1, 2, 6, 4, 8, 7]
```

输入：

lst1 = [3, 6, 1, 8, 1, 9, 2]

lst2 = [3, 1, 2, 6, 4, 8, 7]

输出：

(3, 7)

(6, 4)

(8, 2)

(9, 1)

(2, 8)

答案：

```
1 lst1 = [3, 6, 1, 8, 1, 9, 2]
2 lst2 = [3, 1, 2, 6, 4, 8, 7]
3 for item1 in lst1:
4     for item2 in lst2:
5         if item1 + item2 == 10:
6             print((item1, item2))
7
```

关卡：1、2、3、4、6、7、8

难度：容易

### 38、求满足条件的两位数

$809 * ?? = 800 * ?? + 9 * ??$  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。

809乘以一个两位数 等于 800乘以这个两位数 加上 9乘以这个两位数。而且8乘以这个两位数的结果也是两位数。

输入：无

输出：

$9708 = 809 * 12 = 800 * 12 + 9 * 12$

答案：

```
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)
```

关卡：1、2、3、4、5、6、7

难度：容易

### 39、求0-7，能够组成的奇数的个数

求0—7所能组成的奇数个数，每一位上的数字可以重复，比如33,555都是奇数，最大的数可以是8位数。可以分别计算比如：

组成一位数的奇数有4个

组成二位数的奇数有 $7 * 4$ 个

组成三位数的奇数有 $7 * 8 * 4$

。。。

输入：无

输出：

1位数的奇数有4个

2位数的奇数有28个

3位数的奇数有224个

4位数的奇数有1792个

5位数的奇数有14336个

6位数的奇数有114688个

7位数的奇数有917504个

8位数的奇数有7340032个

sum = 8388608

答案：

```
1 sum = 0
2 s = 0
3 for j in range(8):
```

```

4     if j == 0:
5         s = 4
6     elif j == 1:
7         s = 28
8     else:
9         s *= 8
10    sum += s
11    print("{}位数的奇数有{}个".format(j + 1, s))
12    print('sum = {}'.format(sum))

```

关卡：1、2、3、4、5、6、7

难度：容易

#### 40、寻找偶数

从列表 `lst = [1, 3, 5, 2, 7, 9, 10]` 中寻找第一个出现的偶数并输出。

输入：无

输出：

2

答案：

```

1 lst = [1, 3, 5, 2, 7, 9, 10]
2 for item in lst:
3     if item % 2 == 0:
4         print(item)
5         break

```

关卡：1、2、3、4、5、6、7

难度：容易

#### 41、for循环与while循环嵌套

已知 `lst = [2, 3, 4]`

依次要求用户输入2，3，4的整数倍，先让用户一直输入2的倍数，如果用户输入的正确，输出“输入正确”，否则输出“输入错误”；如果用户输入quit，则停止输入2的倍数，让用户继续一直输入3的倍数，输入3的倍数的过程中，如果用户输入quit，则让用户继续一直输入4的倍数，直到输入quit。

**输入和输出：**如下所示是对应的一组输入与输出：4、8、3、quit、6、quit、8、quit都是从键盘输入的内容。

请输入2的倍数,想停止输入时，输入quit:4

输入正确

请输入2的倍数,想停止输入时，输入quit:8

输入正确

请输入2的倍数,想停止输入时, 输入quit:3

输入错误

请输入2的倍数,想停止输入时, 输入quit:quit

请输入3的倍数,想停止输入时, 输入quit:6

输入正确

请输入3的倍数,想停止输入时, 输入quit:quit

请输入4的倍数,想停止输入时, 输入quit:8

输入正确

请输入4的倍数,想停止输入时, 输入quit:quit

答案:

```
1 lst = [2, 3, 4]
2 for item in lst:
3     while True:
4         input_str = input("请输入{number}的倍数,想停止输入时, 输入
quit:".format(number=item))
5         if input_str == 'quit':
6             break
7         number = int(input_str)
8         if number % item == 0:
9             print("输入正确")
10        else:
11            print("输入错误")
```

关卡: 1、2、3、4、5、6、7

难度: 中等

## 42、列表偏移

lst = [1,2,3,4,5], 列表向右偏移两位后, 变成lst = [4,5,1,2,3]

输入:

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

输出:

[4, 5, 1, 2, 3]

答案:

```
1 lst = [1,2,3,4,5]
2 lst = lst[len(lst)-2:] + lst[:len(lst)-2]
3 print(lst)
```

关卡: 1、2、3、4、5、6、7、8

难度: 容易

### 43、老王卖瓜

老王卖瓜，自卖自夸。第1个顾客来了，买走了他所有西瓜的一半又半个；第2个顾客来了，又买走了他剩余西瓜的一半又半个.....当第9个顾客来时，它已经没有西瓜可以卖了。请问老王原来有多少个西瓜。

输入：无

输出：

老王原有西瓜255个

答案：

```
1 n = 0
2 i = 1
3 while i <= 8:
4     n = (n + 0.5) * 2
5     i = i + 1
6 print('老王原有西瓜{}个'.format(int(n)))
```

关卡：1、2、3、4、5、6、7

难度：容易

### 44、最高与最低的差

输入n个成绩，换行输出n个成绩中最高分数和最低分数的差。

输入：

输入一串字符串为n个成绩（整数表示，范围0~100），以空格隔开。

输出：

一个数字，输出n个成绩中最高分数和最低分数的差。

答案：

```
1 res_str = input("请输入成绩，每个成绩之间使用空格隔开：")
2 res_list = res_str.split(" ")
3 res_list = [int(i) for i in res_list]
4 res_list = sorted(res_list)
5 val = res_list[-1]-res_list[0]
6 print(val)
```

关卡：1、2、3、4、7

难度：容易

### 45、角谷猜想

角谷猜想又称冰雹猜想。它首先流传于美国，不久传到欧洲，后来由一位叫角谷的日本人带到亚洲。因为被称为角谷猜想。

通俗地讲，角谷猜想的内容是这样的：任意给定一个自然数 $n$ ，当 $n$ 是偶数时，将它除以2，即将它变成 $n/2$ ；当 $n$ 是奇数时，就将它变成 $3n+1$ ，.....，若干步后，总会得到1。

**输入：**输入一个自然数字，如：7。

**输出：**

一个数字，输出第一次出现1的位置，如：17。

**答案：**

```
1 n = int(input("请输入一个自然数: "))
2 a = [n]
3 while n != 1:
4     n = n * 3 + 1 if n % 2 else n / 2
5     a.append(n)
6 print(len(a))
```

**关卡：**1、2、3、4、7

**难度：**容易

#### 46、计算分数值

编写一个函数，输入 $n$ 为偶数时，求 $1/2+1/4+\dots+1/n$ ，当输入 $n$ 为奇数时，求 $1/1+1/3+\dots+1/n$

**输入：**

输入一个自然数字，如：4。

**输出：**

一个数字，输出得数：0.75。

**答案：**

```
1 n = int(input('请输入数字:'))
2 s = 0.0
3 if n % 2 == 0:
4     for i in range(2, n + 1, 2):
5         s += 1.0 / i
6 else:
7     for i in range(1, n + 1, 2):
8         s += 1.0 / i
9 print(s)
```

**关卡：**1、2、3、4、7

**难度：**容易



## 47、大写字符串

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。

输入：

输入一个字符串，如："XiaoxiAng"。

输出：

若该字符串全部大写，则输出："该字符串全大写"，并输出该字符串；若该字符串非全部大写，则输出："该字符串非全大写"，并输出该字符串。

答案：

```
1 string = input('请输入一个字符串:')
2 if not string.isupper():
3     print("该字符串非全大写")
4     string = string.upper()
5 else:
6     print("该字符串全大写")
7 print(string)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 48、删除元素

使用一行代码删除一个纯数字列表中小于10的元素。

输入：

空，列表形式如：[11,2,13,14,5,16,17,8]。

输出：

该列表大于等于10的元素组成的列表：[11, 13, 14, 16, 17]。

答案：

```
1 list_1=[11,2,13,14,5,16,17,8]
2 list_2 = [i for i in list_1 if i>=10]
3 print(list_2)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 49、字符串转化为字典

将一个类似于字典形式的字符串转化为一个字典。

输入：

字符串形式如：“小:1,象:2,学:3,院:4”。

输出：

输出生成的字典：{'小': '1', '象': '2', '学': '3', '院': '4'}。

答案：

```
1 str1 = "小:1,象:2,学:3,院:4"
2 def s2d(s):
3     dict_1 = {}
4     for i in s.split(','):
5         key,value = i.split(':')
6         dict_1[key] = value
7     print(dict_1)
8 s2d(str1)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 50、提供类私有化变量接口

将类对象私有化，通过共有方法提供一个读取数据的接口。

输入：

私有化类变量：\_\_fur = "棕色"。

输出：

调用私有化变量的值。

答案：

```
1 class MyCat():
2     __fur = "棕色"
3
4     @property
5     def fur(self):
6         return self.__fur
7
8 tom = MyCat()
9 tom.fur
```

关卡：1、2、3、4、5、6、7、14、15

难度：困难

## 51、字符串转数字

字符串 "123" 转换成 123, 要求不使用如int()的内置接口。

输入:

一串数字形式的字符串: "1234"。

输出:

对应的数字: 1234。

答案:

```
1 s = input("请输入一串数字: ")
2 num = 0
3 for i in s:
4     k = eval(i)
5     num = num * 10 + k
6 print(num)
```

关卡: 1、2、3、4、5、6、7

难度: 中等

## 52、判断密码是否合规

输入密码后判断密码是否包含空格, 包含返回"False", 不包含返回"True"。

输入:

一串字符串形式的密码: "1sdfaA3 4%"。

输出:

返回对应的值。

答案:

```
1 def check_space(pwd):
2     flag = 0
3     for i in pwd:
4         if i.isspace():
5             flag = 1
6             break
7     if flag == 0:
8         return True
9     else:
10        return False
```

关卡: 1、2、3、4、5、6、7

难度: 中等

## 53、输出最大值

读取一个三个数字，使用列表进行排序输出最大值。

输入：

三个数字：1、2、3。

输出：

返回最大值。

答案：

```
1 x = int(input('请输入第一个数:'))
2 y = int(input('请输入第二个数:'))
3 z = int(input('请输入第三个数:'))
4 list_0 = [x,y,z]
5 list_1 = sorted(list_0)
6 print(list_1[-1])
```

关卡：1、2、3、4、5、6、7

难度：简单

## 54、寻找完全数

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 $6=1+2+3$ 。

输入：

两个数字，找出这两个数字之间的所有完全数：1、1000。

输出：

返回对应数字：6, 28, 496。

答案：

```
1 def num(low,high):
2     for i in range(low,high):
3         sum=0
4         a=[]
5         for j in range(1,i):
6             if i%j==0:
7                 sum += j
8                 a.append(j)
9         if sum==i:
10            print("{}次数是完全数，他的因子有:{}".format(i,a))
```

关卡：1、2、3、4、5、6、7

难度：中等

## 55、字符串的和

输入一个数字字符串，返回每一位数字的和。

输入：

一串纯数字字符串：123456。

输出：

返回对应和：21。

答案：

```
1 a = input("请输入一串数字：")
2 sum = 0
3 for i in a:
4     sum += int(i)
5 print(sum)
```

关卡：1、2、3、4、5、6、7

难度：简单

## 56、字符串增长

使用join方法将几个字符串合并起来。

输入：

几个字符串："12"、"3"、"456"。

输出：

返回对应字符串："123456"。

答案：

```
1 str1 = '12'
2 str2 = '3'
3 str3 = '456'
4 strjoin = ''
5 strjoin = strjoin.join([str1,str2,str3])
6 print(strjoin)
```

关卡：1、2、3、4、5、6、7

难度：简单

## 57、数字增加

创建一个将数字作为参数的函数，将数字增加+1并返回结果。

输入：

1个数字：12。

输出:

返回对应字符串: 13。

答案:

```
1 def addi(num):  
2     return num + 1  
3 a = input("请输入一个数字")  
4 print(addi(int(a)))
```

关卡: 1、2、3、4、5、6、7

难度: 简单

## 58、三角面积

编写一个函数, 该函数采用三角形的底和高并返回其面积。

输入:

输入两个数字, 表示三角形的底和高: 12, 4。

输出:

返回对应的面积: 24。

答案:

```
1 def tArea(base, height):  
2     return 1/2*base*height  
3 a = float(input("请输入三角形的底边长度: "))  
5 b = float(input("请输入三角形的高的长度: "))  
6 print(tArea(a,b))
```

关卡: 1、2、3、4、5、6、7

难度: 简单

## 59、汉诺塔问题

现在有n个圆盘从上往下从小到大叠在第一根柱子上, 要把这些圆盘全部移动到第三根柱子要怎么移动, 汉诺塔问题的以下几个限制条件:

1. 在小圆盘上不能放大圆盘。
2. 在三根柱子之间一回只能移动一个圆盘。
3. 只能移动在最顶端的圆盘。

输入:

请输入圆盘的起始个数: 4

输出:

移动A的一个圆盘到B  
移动A的一个圆盘到C  
移动B的一个圆盘到C  
移动A的一个圆盘到B  
移动C的一个圆盘到A  
移动C的一个圆盘到B  
移动A的一个圆盘到B  
移动A的一个圆盘到C  
移动B的一个圆盘到C  
移动B的一个圆盘到A  
移动C的一个圆盘到A  
移动B的一个圆盘到C  
移动A的一个圆盘到B  
移动A的一个圆盘到C  
移动B的一个圆盘到C

答案:

```
1 def move(n, a = 'A', b = 'B', c = 'C'):  
2     if n == 1:         #递归条件当n为1,时,执行移动的操作  
3         print('移动{}的一个圆盘到{}'.format(a, c))    #打印移动方法  
4         return  
5     move(n-1,a,c,b)    #先把n-1个盘子,从a移动到b  
6     move(1,a,b,c)      #再将剩下的1个盘子,从a移动到c  
7     move(n-1,b,a,c)    #柱子b上面有n-1个盘子,再将盘子从b,借助a,移动到c  
10 floor = int(input("请输入圆盘层数"))  
11 move(floor)
```

关卡: 1、2、3、4、5、6、7、8、9、10

难度: 困难

## 60、字典生成问题

使用给定的整数n, 编写一个程序生成一个包含(i, i\*i)的字典, 该字典包含1到n之间的整数(两者都包含)。然后程序应该打印字典。

输入:

请输入: 6

输出:

{1:1, 2:4, 3:9, 4:16, 5:25, 6:36}

答案:

```
1 n=int(input('请输入一个数字: '))  
2 d=dict()  
3 for i in range(1,n+1):  
4     d[i]=i*i  
5
```

6 print(d)

关卡：1、2、3、4、5、6、7、8

难度：简单

### 61、比较大小问题

给出两个数字M、N，比较M+N、M-N、M\*N、M/N、M^N五种运算的大小，并将最大的数输出。

输入：

请输入：6，0.2

输出：

五个数字的最大值：30.0

答案：

```
1 M = float(input("请输入M: "))
2 N = float(input("请输入N: "))
3 a = {}
4 a["M+N"] = M+N
5 a["M-N"] = M-N
6 a["M*N"] = M*N
7 a["M/N"] = M/N
8 a["M^N"] = M**N
9 keys = list(a.values())
10 keys.sort()
11 print("最大值是: ", keys[-1])
```

关卡：1、2、3、4、5、6、7、8

难度：简单

### 62、python的增长

给定一个数字n，返回“python”对应的n个o的字符串，字符串变化方式如下：

- a、字符串中o的数目应和n相等，除非n小于1（在这种情况下，请返回“python”）。
- b、如果2能整除n，请在末尾添加一个感叹号。
- c、如果5能整除n，请将所有字符串字母大写。
- d、如果2和5都能整除n，请将所有字符串字母大写并在结尾添加一个感叹号。

输入：

请输入数字：2

请输入数字：5

请输入数字：10

输出：



输出: pythoon!

输出: PYTHOOOOON

输出: PYTHOOOOOOOOOOON!

答案:

```
1 def inten(n):
2     if n < 1:
3         return 'python'
4     s = 'pyth{}n'.format('o'*n)
5     if not n%2:
6         s += '!'
7     if not n%5:
8         s = s.upper()
9     return s
10 n = int(input("请输入一个数字"))
11 print(inten(n))
```

关卡: 1、2、3、4、5、6、7

难度: 中等

### 63、字典键的排序

使用一个匿名函数, 使列表按照一个字典的键为keys的值从大到小排序。

输入:

```
lst = [{'keys':1, 'value':3},{'keys':3,'value':1},{'keys':5,'value':1},{'keys':2,'value':1}]
```

输出:

```
[{'keys': 5, 'value': 1}, {'keys': 3, 'value': 1}, {'keys': 2, 'value': 1}, {'keys': 1, 'value': 3}]
```

答案:

```
1 lst = [{'keys':1, 'value':3},{'keys':3,'value':1},{'keys':5,'value':1},
2         {'keys':2,'value':1}]
3 func = lambda x:x.get('keys')
5 print(sorted(lst,key = func,reverse=True))
```

关卡: 1、2、3、4、5、6、7

难度: 简单

### 64、匿名函数的列表操作

使用一个匿名函数, 使列表中小于10的元素全部剔除。

输入:

```
list_0 = [-1, 42, 66, 3, -32, 100, 5]
```

输出:

```
[42, 66, 100]
```

答案:

```
1 list_0 = [-1, 42, 66, 3, -32, 100, 5]
2 list_1 = list(filter(lambda i:i >= 10, list_0))
4 print(list_1)
```

关卡: 1、2、3、4、5、6、7

难度: 中等

### 65、自动生成字符串

假设有三个列表: a = ['小马', '小羊', '小鹿'], b = ['草地上', '电影院', '家里'], c = ['看电影', '听故事', '吃晚饭']。试编写程序, 随机生成三个0-2范围内的整数, 将其作为索引分别访问三个列表中的对应元素, 然后进行造句。例如, 随机生成三个整数分别为1, 0, 2, 则输出句子“小羊在草地上吃晚饭”。

输出:

```
小马草地上看电影
```

答案:

```
1 import random
2 a = ["小马", "小羊", "小鹿"]
3 b = ["草地上", "电影院", "家里"]
4 c = ['看电影', '听故事', '吃晚饭']
5 L = [random.randint(0,2) for _ in range(3)]
6 print(a[L[0]]+b[L[1]]+c[L[2]])
```

关卡: 1、2、3、4、5、6、7

难度: 中等

### 66、自动分裂list

现在给定一个list: a = [1,2,3,4,5,6], 要求使用匿名函数将个list根据奇数偶数分成两个list。

输出:

```
[1, 3, 5]
```

```
[2, 4, 6]
```

答案:

```
1 a = [1,2,3,4,5,6]
2 b = list(filter(lambda i:i%2 == 1, a))
```

```

3 c = list(filter(lambda i:i%2 == 0, a))
4 print(b)
5 print(c)

```

关卡：1、2、3、4、5、6、7

难度：中等

## 67、来自农场主的求助

在这个挑战中，一位农民要您帮忙统计他所有动物有多少条腿。农民养殖了三种动物：鸡两条腿、牛四条腿、猪四条腿。农夫已经数了他的动物，他给您每种物种的小计。您必须实现一个函数，该函数返回所有动物的腿的总数。

输入：

4 2 5

分别代表四只鸡、两头牛、五只猪

输出：

36

答案：

```

1 a = input("请输入动物的个数，用空格隔开： ")
2 a = a.split(" ")
3 animals = lambda x, y, z: int(x) * 2 + int(y) * 4 + int(z) * 4
4 print(animals(a[0], a[1], a[2]))

```

关卡：1、2、3、4、5、6、7

难度：中等

## 68、切分字符串（文慧，弄成选作吧，逻辑有点难）

编写一个函数判断一个纯数字字符串是否能拆成几个连续加1的数字的形式。例如：2345这个字符串可以拆成2、3、4、5这几个数的拼接，232425这个字符串可以拆成23、24、25这几个数字的拼接。而232这个字符串无论如何拆解不成几个数字的组合

拆解规则：

- 1.数字必须是连续的，356这样是不合规的。
- 2.数字必须是增加的，432同样是不合规的。
- 3.必须能拆成两个数字或以上，1、2、3等这样单个位数字字符串不合规。
- 4.注意可以跨位数，例如910，这个数可以拆成9、10。该条件选做！

输入及输出：

232425	True
444445	True
1234567	True
123412351236	True

57585960616263	True
500001500002500000	True
919920921	True
2324256	False
1235	False
121316	False
12131213	False
54321	False
56555453	False
90090190290	False
35236237238	False
910	True

### 答案：

答案思路是将每个字符串的第一位、第一到二位、....依次取出，直到位数超过字符串的一半即停止。将取出的数字成一个满足条件的字符串，并判断我们生成的字符串和原字符串是否相等。例如现在输入的是235，我们先取第一位2，用2连续加1，生成一个列表[2,3,4]，用这个列表生成一个字符串234，发现与235这个字符串不相等，再取前两位23，生成一个满足条件的连续的list：[23, 24]，生成字符串2324，取前三位字符串232，同样与235不相等。说明这个字符串不能被拆解。

后面的循环是为了判断232这种两位数的情况，如果最后一位增加进来的数字不完整，即23、24的24只增加进来了2，那么这个字符串再这种切分条件下一定不行。

```

1 def ascending(txt):
2     if len(txt) < 2:
3         return False
4     for i in range(1, len(txt) // 2 + 1):
5         # a是我们生成的list, 【2、3、4】，【23, 24】。
6         # -(-len(txt)//i))表示正数的向上取整
7         a = list(range(int(txt[:i]), int(txt[:i]) - (-len(txt)//i)))
8         b = "" #初始话一下我们要生成的字符串
9         length = 0 #初始化以下字符串长度，这个长度的含义是现在已有的字符和即将加进
          来的字符串合起来的总长度。
10        for j in a: #这里是依次将a的元素添加进来
11            length += len(str(j)) #统计加进来j之后生成字符串的长度
12            if length < len(txt): #当生成的字符串长度小于原字符串，继续添加
13                b = b + (str(j))
14            # 当生成的字符串长度等于原字符串，继续添加
15            # 添加完成后break这个字符串就可以进行比较了。
16            elif length == len(txt):
17                b = b + (str(j))
18                break
19            # 当即将生成字符串大于原字符串的时候，例如【23, 24】
20            # 24实际只取到了2，一定不满足，退出。
21            elif length > len(txt):
22                break

```

```
23     # 进行比较，满足则返回True
24     if b == txt:
25         return True
26     #当循环都执行完成后，没有满足的条件出现，说明是不满足的。返回False。
27     return False
```

关卡：1、2、3、4、5、6、7

难度：困难

## 69、真值检查

创建一个函数，该函数返回一个只有True和False的列表中True的值数。

输入：

请输入list：[True, False, False, True, False]

输出：

2

答案：

```
1 def count_true(lst):
2     return sum(lst)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 70、有利润的交易

我们假定再一次交易中我们投入的成本和产出的关系是：成本 \* 比率 = 产出，而买方会提供一个价格，当这个价格小于产出时，我们将决绝这次交易。输入为三个数字，分别代表成本、比率、买方出价。如果交易成立则打印：成功交易；如果交易不成立则打印：拒绝交易。

输入：

请输入成本：10

请输入比率：1.2

请输入买方价格：13

输出：

成功交易

答案：

```
1 a = float(input("请输入成本："))
2 b = float(input("请输入比率："))
```

```
3 c = float(input("请输入买方价格: "))
5 if a * b <= c:
6     print("成功交易")
7 else:
8     print("拒绝交易")
```

关卡：1、2、3、4、5、6、7

难度：容易

## 71、列表的生长

每次从键盘上读取一个列表，将这个列表添加进之前已有的列表，直到从键盘上读入0。读入0后，输出最后的列表并结束。

输入：

请输入数组(输入0退出): [1,2]

请输入数组(输入0退出): [4,5]

请输入数组(输入0退出): 0

输出：

[1, 2, 4, 5]

答案：

```
1 b = []
2 while True:
3     a = eval(input("请输入数组(输入0退出): "))
4     if a == 0:
5         print(b)
6         break
7     else:
8         b.extend(a)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 72、保留重复项

在一个全是字符串的列表中，保留那些只有重复相同字符的字符串。例如：“111”则是具有重复相同字符的字符串，“xyz”则不具有重复相同字符。具有单个字符的字符串如“3”也被视为具有重复相同字符的字符串。如果没有重复相同字符的字符串，则返回一个空数组。

输入：

["aa", "123", "444", "b", "xyz"]

["ann", "123", "4g4", "de", "xyz"]

输出:

['aa', '444', 'b']

[]

答案:

```
1 def identical_filter(lst):  
2     return [i for i in lst if len(set(i))==1]
```

关卡: 1、2、3、4、5、6、7

难度: 容易

### 73、数字判断

给定一个int类型的数字, 这个数字的每一位的加和如果是奇数则输出: 奇数, 如果各个位的加和是偶数, 那么输出: 偶数。例如123,  $1+2+3=6$ , 输出: 偶数。

输入:

1

121

输出:

奇数

偶数

答案:

```
1 def dec(num):  
2     result = sum([int(x) for x in str(num)])  
3     if result%2 == 0:  
4         return ("偶数")  
5     else:  
6         return ("奇数")
```

关卡: 1、2、3、4、5、6、7

难度: 容易

### 74、找最接近的整倍数。

给定两个值, m和n, n始终为正整数。要求返回一个最接近m的n的整倍数, 如果m正巧处于两个整倍数的中间, 则返回较大的整倍数。

输入:

(47, 7)

输出:

答案:

```
1 def ron(num, n):  
2     min = [num - i for i in range(n) if (num - i) % n == 0][0]  
3     max = [num + i for i in range(n) if (num + i) % n == 0][0]  
4     return min if num - min < max - num else max
```

关卡: 1、2、3、4、5、6、7

难度: 容易

### 75、列表中的元素个数

在一个嵌套的列表中我们想知道这个列表共有多少个非嵌套的元素。例如[1, [2, [[3], 4], 5], 6], 一共有6个数字, 即有6个非嵌套的元素。

输入:

```
[1, [2, [[3], 4], 5], 6]  
[]
```

输出:

```
6  
0
```

答案:

```
1 def gle(lst):  
2     return 1 if type(lst) is int else sum(gle(i) for i in lst)
```

关卡: 1、2、3、4、5、6、7

难度: 容易

### 76、出现顺序的比较

编写一个函数, 实现以下功能。给出一个全部为小写字母的字符串, 并给出两个在字符串中一定存在的字母a和b。当a字母在字符串中最后一次出现的位置比b字母第一次出现的位置还要靠前时, 则返回True, 否则返回False。

输入:

```
"xiao xiang xue yuan xue xi python", "a", "n"  
"happy birthday", "p", "b"
```

输出:

```
False  
True
```



答案:

```
1 def f_and_s(s, a, b):  
2     return s.rindex(a) < s.index(b)
```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 77、数字的和谐

编写一个接受一串int类型数字, 并返回一个一位数字的函数, 这个一位数是对输入数字的所有位数字求和的结果。当总和不是一位数时, 重复求和, 直到获得一位数字为止。例如129,  $1+2+9=12$ ,  $1+2=3$ 。输出3

输入:

987

输出:

6

答案:

```
1 def dec(num):  
2     while num > 9:  
3         num = sum(int(x) for x in str(num))  
4     return num
```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 78、简化分数

创建一个函数, 返回传入分数的最简分数。

输入:

"2/12"

输出:

"1/6"

答案:

```
1 def sim(num):  
2     a, b = map(int, num.split('/'))  
3     if a % b == 0:  
4         return str(a//b)  
5     for i in range(min(a, b), 1, -1):  
6         if a % i == 0 and b % i == 0:
```

```
7         return '{}/{ {}'.format(a//i, b//i)
8     return num
9     sim("2/12")
```

关卡：1、2、3、4、5、6、7

难度：容易

## 79、重量平衡的字符串

我们根据字母在字母表中的顺序为每一个字符进行编号，a是数字1，b是数字2，以此类推，z是数字26。给出一个纯小写字母的字符串，我们从字符串的中间将字符串分成两半，奇数个数的字符串，中间字母将被忽略。通过计算，左半边字符串的数字和右半边的数字和。若这两者相等，则打印：相等，否则打印：不相等。回文字符串始终相等。

输入：

python  
abba

输出：

不相等  
相等

答案：

```
1 word = input("请输入字符串：")
2 if sum([ord(i)-96 for i in word[:len(word)//2]]) == sum([ord(i)-96 for i
3   in word[-(len(word)//2):]]):
4     print("相等")
5 else:
6     print("不相等")
```

关卡：1、2、3、4、5、6、7

难度：容易

## 80、正交向量

创建一个判断两个向量是否正交的函数，这两个向量的储存格式为一维数组。如果两个向量a和b的点积等于零，则它们是正交的。

输入：

[1, 2], [2, -1]  
[1, 2, 0], [2, -1, 5]  
[6, -1], [6, 7]

输出：

正交  
正交

非正交

答案:

```
1 A = eval(input("请输入向量A: "))
2 B = eval(input("请输入向量B: "))
3 if not sum(x*y for x, y in zip(A,B)):
4     print("正交")
5 else:
6     print("非正交")
```

关卡: 1、2、3、4、5、6、7

难度: 容易

### 81、阶乘的递归

编写一个函数，用来实现一个值的阶乘。（注意：阶乘不要写的太大，否则会导致电脑卡死。）

输入:

3

输出:

6

答案:

```
1 def fac(n):
2     return n * fac(n - 1) if n > 1 else 1
```

关卡: 1、2、3、4、5、6、7

难度: 容易

### 82、给客人打招呼

编写一个迎宾机器人，在客人进门时对每一位客人说：Hello。传入的是一个客人名字的列表，输出格式为：“Hello, xxx, Hello, xxx。...”

输入:

["Green", "Black", "White"]

输出:

Hello Green, Hello Black, Hello White

答案:

```
1 names = eval(input("请输入客人列表: "))
2 print(", ".join(["Hello " + n for n in names]))
```

关卡：1、2、3、4、5、6、7

难度：容易

### 83、汉明距离

汉明距离指的是两个字符串在对应位置字母不同的个数，例如：“abcd”和“abcb”的汉明距离是1，“abcd”和“bcde”的汉明距离是4，创建一个函数来完成这个计算。

输入：

“abcd”，“bcde”

输出：

4

答案：

```
1 def hamming(A, B):  
2     return sum(x!=y for (x,y) in zip(A, B))
```

关卡：1、2、3、4、5、6、7

难度：容易

### 84、计算反弹高度

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？

输入：

请输入起始高度：100.0

请输入反弹次数：10

输出：

总高度：tour = 299.609375

第10次反弹高度：height = 0.09765625

答案：

```
1 tour = []  
2 height = []  
3  
4 hei = float(input("请输入起始高度：")) # 起始高度  
5 tim = int(input("请输入反弹次数：")) # 次数  
6  
7 for i in range(1, tim + 1):  
8     # 从第二次开始，落地时的距离应该是反弹高度乘以2（弹到最高点再落下）
```

```

9     if i == 1:
10         tour.append(hei)
11     else:
12         tour.append(2*hei)
13     hei /= 2
14     height.append(hei)
15
16 print('总高度: tour = {0}'.format(sum(tour)))
17 print('第10次反弹高度: height = {0}'.format(height[-1]))

```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 85、数据加密

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(如果加上5后的结果大于等于10，那么只保留这个结果的个位数),然后将第一位和第四位交换，第二位和第三位交换。

输入：

输入四个数字：5678

输出：

3210

答案：

```

1 a = int(input('输入四个数字:'))
2 # 将四位数的千位、百位、十位、个位分别保存到列表中
3
4 aa = []
5 aa.append(a // 1000)      # 千位
6 aa.append(a % 1000 // 100) # 百位
7 aa.append(a % 100 // 10)   # 十位
8 aa.append(a % 10)          # 个位
9
10 for i in range(4):
11     aa[i] += 5
12     aa[i] %= 10
13 aa[0],aa[3] = aa[3],aa[0]
14 aa[1],aa[2] = aa[2],aa[1]
15 for i in aa:
16     print(i, end=" ")

```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 86、字符统计

输入一行字符，分别统计出其中的英文字母、空格、数字和其它字符的个数。

输入：

输入四个数字：123xiaoxiang xueyuan235\*(dfl

输出：

char = 19,space = 2,digit = 6,others = 2

答案：

```
1 s = input('请输入一个字符串:')
2 letters = 0
3 space = 0
4 digit = 0
5 others = 0
6 i=0
7 while i < len(s):
8     c = s[i]
9     i += 1
10    if c.isalpha():
11        letters += 1
12    elif c.isspace():
13        space += 1
14    elif c.isdigit():
15        digit += 1
16    else:
17        others += 1
18 print('char = {},space = {},digit = {},others = {}'.format(letters,space,digit,others))
```

关卡：1、2、3、4、5、6、7

难度：容易

## 87、反向输出

按相反的顺序输出列表的值：通过命令行运行python程序，并传入三个元素，将这三个元素保存到一个列表中，然后反向输出列表中的元素。

输入：

在命令行中输入：

python 文件名.py 1 2 3

输出：

3 2 1

答案：

```
1 import sys
2 a = []
3 a.append(sys.argv[1])
4 a.append(sys.argv[2])
5 a.append(sys.argv[3])
```

```
6 for i in a[::-1]:
8     print(i, end=" ")
```

关卡：1、2、3、4、5、6、7

难度：容易

## 88、匿名函数

创建一个匿名函数，这个函数的作用是返回两个数中较大的那个数。

输入：

10 20

输出：

20

答案：

```
1 # 定义一个匿名函数：返回两个数中的较大数
2 MAX = lambda x,y : (x > y) * x + (x < y) * y
3 a = float(input("请输入一个数："))
5 b = float(input("请再输入一个数："))
6 # 调用函数并打印结果
8 print(MAX(a,b))
```

关卡：1、2、3、4、5、6、7

难度：容易

## 89、统计子串出现次数

计算字符串中子串出现的次数：输入一个字符串a和一个子字符串b，请编写程序统计子字符串b在字符串a中出现的次数。

输入：

请输入一个字符串：xiaoxiangxueyuan

请输入一个子字符串：xi

输出：

子字符串出现的次数为：2

答案：

```
1 a = input('请输入一个字符串:')
2 b = input('请输入一个子字符串:')
3 n = a.count(b)
4 print("子字符串出现的次数为：", n)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 90、递归求阶乘

请使用递归的方法计算一个整数的阶乘。

输入：

请输入一个整数：5

输出：

120

答案：

```
1 def fact(j):
2     sum = 0
3     if j == 0:
4         sum = 1
5     else:
6         sum = j * fact(j - 1)
7     return sum
8
9 n = int(input("请输入一个整数: "))
10 print(fact(n))
```

关卡：1、2、3、4、5、6、7

难度：容易

## 91、生成字典

使用给定的整数n，编写一个程序生成一个包含(i, i\*i)键值对的字典。

输入：

请输入一个数字：8

输出：

{1:1, 2:4, 3:9, 4:16, 5:25, 6:36, 7:49, 8:64}

答案：

```
1 n=int(input('请输入一个数字: '))
2 d=dict()
3 for i in range(1,n+1):
4     d[i]=i*i
5
6 print(d)
```

关卡：1、2、3、4、5、6、7

难度：容易

## 92、计算pi的值



利用下列公式计算 $\pi$ 的近似值，直到最后一项的绝对值小于 $10^{-5}$ 为止。

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

答案：

```
1 n=0.0
2 p=1
3 while p<=100000:
4     if p%4==1:
5         n=n+1/p
6     else:
7         n=n-1/p
8     p=p+2
9 else:
10    print (n*4)
```

关卡：1、2、3、4、5、6、7

难度：容易

### 93、打印不重复单词

请编写一个程序，从键盘接收一系列空格分隔的单词作为输入，然后删除所有重复的单词，最后按字母排序后打印这些单词。

输入：

hello world and practice makes perfect and hello world again

输出：

again and hello makes perfect practice world

答案：

```
2 s = input('请输入一组字符串：')
3 words = [word for word in s.split(" ")]
4 print (" ".join(sorted(list(set(words)))))
```

关卡：1、2、3、4、5、6、7

难度：容易

### 94、是否被5整除

请编写一个程序，从键盘接收一系列逗号分隔的4位二进制数作为输入，然后检查它们是否可被5整除，将可被5整除的数字将以逗号分隔的顺序打印。

输入：

0100,0011,1010,1001

输出：

答案:

```
1 value = []
2 num = input('请输入逗号分隔的4位二进制数: ')
3 items=[x for x in num.split(',')]
4 for p in items:
5     intp = int(p, 2)
6     if not intp%5:
7         value.append(p)
8
9
10 print (' ','.join(value))
```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 95、求和函数

请编写一个求和函数，函数可以接收任意多个整数，函数返回所有整数的平方和。

答案:

```
1 def square_sum(*param):
2     sum = 0
3     for i in param:
4         sum += i**2
5     return sum
6 print(square_sum(1,2,3))
7 print(square_sum(1,2,3,4))
```

关卡: 1、2、3、4、5、6、7

难度: 容易

## 96、对字典进行排序

假设现在有一个字典 `d = {'a':24,'g':52,'i':12,'k':33}`，请编写程序实现对字典按value值进行排序。

输入:

```
d = {'a':24,'g':52,'i':12,'k':33}
```

输出:

```
[('i', 12), ('a', 24), ('k', 33), ('g', 52)]
```

答案:

```
1 d = {'a':24,'g':52,'i':12,'k':33}
2 print(sorted(d.items(),key=lambda x:x[1]))
```

关卡：1、2、3、4、5、6、7

难度：容易

## 97、分段函数

请编写一个函数用来计算下面这个分段函数的值：

$$f(x) = \begin{cases} -1; & |x| \geq 300 \\ \frac{x^3}{\lg(|x|+2.6)}; & |x| < 300 \end{cases}$$

输入：

301

输出：

-1

答案：

```
1 import math
2 def f(x):
3     if abs(x) >= 300:
4         return -1
5     else:
6         return x**3/math.log(abs(x)+2.6,10)
7 x = float(input("请输入一个数字："))
8 print(f(x))
```

关卡：1、2、3、4、5、6、7

难度：容易

## 98、复制列表

请编写程序实现将一个列表a的内容复制到另一个列表b中。

要求：对列表a中元素的修改不会影响列表b中的元素。

答案：

```
1 # 方法1:
2 a = [1,2,3,4]
3 b = a[:]
```

```

5 b[0] = 100
6 print(a, b)
8 # 方法2:
9 import copy
10 a = [1,2,3,4]
11 b = copy.copy(a)
12 b[0] = 100
14 print(a, b)

```

关卡：1、2、3、4、5、6、7

难度：容易

## 99、日志中查找文件名

请编写程序实现在日志中查找指定后缀的文件名。

例如，一个日志字符串如下所示：

```
"192.168.88.1 -- [30/Jul/2017:12:54:40 +0800] "GET /MyDemoWeb/hadoop.jsp
HTTP/1.1" 200 242"
```

请找出后缀为jsp的文件，并打印出文件名：

hadoop.jsp

答案：

```

1 # 日志字符串
2 s = '192.168.88.1 -- [30/Jul/2017:12:54:40 +0800] "GET /MyDemoWeb/hadoop.
   jsp HTTP/1.1" 200 242'
3 # 使用find方法返回'.'的索引位置，+4可以定位到'p'后一个的索引位置
4 end = s.find('.jsp') + 4
5 # 使用rfind方法在[0,end)区间查找'/',+1可以查找到'h'
6 begin = s.rfind('/', 0, end) + 1
8 print(s[begin:end])
10

```

关卡：1、2、3、4、5、6、7

难度：容易

## 100、冒泡排序

请编写程序实现冒泡排序。

冒泡排序（Bubble Sort）是一种简单的排序算法。它重复地遍历要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。

答案：

```

1 def bubble_sort(list):
2     n = len(list)
3     for i in range(n - 1):
4         for j in range(0, n - 1 - i):
5             if list[j] > list[j + 1]:
6                 list[j], list[j + 1] = list[j + 1], list[j]

```

```
7  
8 a = [3,5,2,7,1]  
9 bubble_sort(a)  
10 print(a)
```

关卡：1、2、3、4、5、6、7

难度：容易