Python一本通

信息技术学考教程

第四章　循环结构

前面我们学习了顺序结构与分支结构的程序设计。在实际应用中，会经常遇到许多有规律性的重复运算，这就需要掌握循环结构程序设计。

Python语言提供三种循环结构:for、while和循环嵌套。

第一节　for循环

在编写程序时，一般用循环结构来处理重复执行某一条或某一组语句的问题。在Python语言中，根据循环的描述方法不同，循环结构可分为两种类型：一种是**计数循环(for循环)**，即提前预定循环体重复执行的次数；另一种是**条件循环(while循环)**，即当满足一定条件时反复执行循环体内的语句。

本节主要研究for循环。

一、语句格式

```
for <循环变量> in <序列>:
    <循环体>
```

【说明】

(1)<序列>通常是字符串、列表、range对象实例等，当序列中的元素全部遍历完成后，程序就会自动退出循环。

(2)for语句每次从序列中取出一个元素赋值给循环变量(循环变量初值即为序列中的第一个元素值)，当依次访问完序列中所有元素后，循环结束。需要注意的是，for…in后面的冒号“:”不能省略。循环结束后，变量的值停留在序列的最后一个内容上。

(3)冒号“:”代表下一行是循环体的第一行，整个循环体在书写时需要缩进，Python惯例缩进4个空格，缩进的语句块都是需要重复执行的部分。必须遵守缩进要求，否则，循环结构不能正常运行。

二、例题应用

例4.1　输出连续10个数浙版

【题目描述】

从0开始输出，连续输出10个数。

【输入格式】

(无)

【输出格式】

输出10个数，每个数中间有1个空格。

【样例输入】

(无)

【样例输出】

0 1 2 3 4 5 6 7 8 9

【题目分析】

若序列中的元素为有序整数，则可利用内建函数range来实现。

【代码实现】

```
for num in range(0,10,1):
    print(num,end=' ')
```

【说明】

(1)range函数创建一个整数序列，常用for循环，语法为：range(初值,终值[,步长])，计数从初值开始，到终值（不包括终值）结束。若初值省略默认值为0。步长是序列中的每个元素之间的差，若步长省略时默认值为1。range(0,10,1)可写成range(10)。

(2)for循环执行时，循环变量依次从序列中取出相应的元素。思考一下：程序中步长1改为2，输出几个数？是哪几个数？

(3)代码print(num,end=' ')中的end=' '表示输出当前的内容后，在末尾加一个空格，不换行接着输出下一个print()的输出内容。如果缺省的话，print()默认值是\n换行符作为其结束值。

例4.2　网购笔记本[粤教]

【题目描述】

某小组决定网购单价为1元至10元的十种笔记本各一本，求一共要花多少钱？

【输入格式】

（无）

【输出格式】

一行一个整数，表示答案。

【题目分析】

程序需要有一个循环变量从1变化到10，将该变量设为i，i每次加到累加和sum中；变量i每增加1时，就和变量sum进行一次加法运算，变量sum记录的是累加的结果。

【代码实现】

```
# sum储存价格之和，初始值赋为0
sum=0
# 枚举i是1~10之间的数
for i in range(1,11):
    sum=sum+i
print(sum)
```

例4.3　温度对应表1沪版

【题目描述】

计算华氏100度到105度所对应的摄氏温度。精确到小数点后2位。

换算公式 C=5×(F-32)÷9(其中C表示摄氏温度，F表示华氏温度)。

【输入格式】

（无）

【输出格式】

若干行每行两个数，表示华氏温度(整数，占8个字符宽度)与摄氏温度(占10个字符宽度，保留两位小数)。

【样例输入】

（无）

【样例输出】

```
100     37.78
101     38.33
102     38.89
103     39.44
104     40.00
105     40.56
```

【题目分析】

for循环枚举摄氏温度，并运用公式换算即可。

【代码实现】

```
for f in range(100,106):
    c=5*(f-32)/9
    print('%8.0f%10.2f'%(f,c))
```

例4.4　温度对应表2沪版

【题目描述】

人体温度计的合理范围是华氏90度到110度。给定华氏温度的下限 l，上限 h，输出此范围的华氏摄氏温度对应表。

【输入格式】

第一行整数 l。

第二行整数 h。

【输出格式】

若干行每行两个数，表示华氏温度(整数，占8个字符宽度)与摄氏温度(占10个字符宽度，保留两位小数)。

如果下限 l 大于等于上限 h，输出“输入的下限应该小于上限”。

【样例输入】

```
93
105
```

【样例输出】

```
      93      33.89
      94      34.44
      95      35.00
      96      35.56
      97      36.11
      98      36.67
      99      37.22
     100      37.78
     101      38.33
     102      38.89
     103      39.44
     104      40.00
```

例4.4　温度对应表2沪版

【题目描述】

人体温度计的合理范围是华氏90度到110度。给定华氏温度的下限 l，上限 h，输出此范围的华氏摄氏温度对应表。

【输入格式】

第一行整数 l。

第二行整数 h。

【输出格式】

若干行每行两个数，表示华氏温度(整数，占8个字符宽度)与摄氏温度(占10个字符宽度，保留两位小数)。

如果下限 l 大于等于上限 h，输出“输入的下限应该小于上限”。

【题目分析】

特判不合法情况，其余for循环枚举华氏温度，并运用公式换算即可。

【代码实现】

```
l=int(input()) # 输入华氏温度的下限 l
h=int(input()) # 输入华氏温度的上限 h
if l>=h:
    print("输入的下限应该小于上限")
else:
    for f in range(l,h):
        c=5*(f-32)/9
        print('%8.0f%10.2f'%(f,c))
```

例4.5　沃利斯公式 沪版

【题目描述】

使用沃利斯公式求圆周率 π。

沃利斯公式：

$$\frac{\pi}{2}=\frac{2}{1}\times\frac{2}{3}\times\frac{4}{3}\times\frac{4}{5}\times\frac{6}{5}\times\frac{6}{7}\times\frac{8}{7}\times\frac{8}{9}\times\cdots$$

【输入格式】

（无）

【输出格式】

一行一个浮点数，表示圆周率 π。

精确到小数点后4位。

【样例输入】

（无）

【样例输出】

3.1415

【题目分析】

套用题中公式，当右式枚举至10000项时，累乘项足够小，右式可约等于左式。

【代码实现】

```
s=1
# 枚举i是2~20000的数，枚举的步长是2
for i in range(2,20000,2):
    # item=i/(i-1)*i/(i+1)，写作i*i/(i*i-1)
    item=i*i/(i*i-1)
s=s*item
pi=2*s
# 精确到小数点后4位
print('%.4f'%(pi))
```

【课堂练习】

练4.1　for循环求和

【题目描述】

利用for循环。计算输出1+2+3+...+n的和。

【输入格式】

输入n。

【输出格式】

如题述，之和。

【输入样例】

10

【输出样例】

55

【代码实现】

```
n=int(input()) # 输入给定1个整数n
sum=0

# i初始值1，终值为n，每次增量为1
for i in range(1,n+1):
    sum+=i
print(sum)
```

练4.2　输出偶数

【题目描述】

按照由小到大的顺序，输出1~n之间的所有偶数。

【输入格式】

输入n。

【输出格式】

输出为一行，各个偶数之间用一个空格隔开。

【输入样例】

10

【输出样例】

2 4 6 8 10

【题目分析1】

对于1~n之间的n个数字i，直接重复进行判断，如果i是偶数，则输出i的值。

【代码实现1】

```
n=int(input()) # 输入给定1个整数n
sum=0
# 对于i取1至n之间的每一个整数，都重复操作
for i in range(1,n+1):
    # 如果i为偶数，则输出i的值
    if i%2==0:
        # end=" "表示输出之间用“ ”隔开
        print(i,end=" ")
```

程序将1-n之间的所有数字都列举出来，然后一一判断，符合偶数条件的，就输出。这种思想，本质上是穷举。穷举法保证在求解的过程中，所有可能解都会判断到，不会丢解，缺点就是效率不高。

练4.2　输出偶数

【题目描述】

按照由小到大的顺序，输出1~n之间的所有偶数。

【输入格式】

输入n。

【输出格式】

输出为一行，各个偶数之间用一个空格隔开。

【输入样例】

10

【输出样例】

2 4 6 8 10

【题目分析2】

在上述分析的基础上，再进一步分析：我们都知道，相邻偶数之间的差值为2，所以，我们还可以设置变量的初值为2，增量为2的for循环，使得循环次数减少为n/2次。

【代码实现2】

```
n=int(input()) # 输入给定1个整数n
sum=0

# i初始值2，终值为n，每次增量为2
for i in range(2,n+1,2):
    print(i,end=" ")
```

练4.3　输出奇偶数之和

【题目描述】

利用for循环，分别输出1~n之间的所有奇数的和、偶数的和。

【输入格式】

输入n。

【输出格式】

输出为一行，两个数(用一个空格隔开)，偶数之和与奇数之和。

【输入样例】

10

【输出样例】

30 25

【代码实现】

```
n=int(input()) # 输入给定1个整数n

#sum1、sum2分别存放偶数和、奇数和，均初始化为0
sum1=0
sum2=0

#对于i取1至n之间的每一个整数，都重复操作
for i in range(1,n+1):
    if i%2==0:     # 偶数累加到sum1中
        sum1+=i
    else:          # 奇数累加到sum2中
        sum2+=i

# 输出偶数和、奇数和
print(sum1,sum2)
```

【题目分析】

很容易找到所有的偶数和奇数，继而计算其和。假设用变量sum1和sum2分别存放偶数与奇数和，累加就是在sum1或sum2的基础上，加上一个数字，改变累加变量的值；再加上一个数字，改变累加变量的值；.....；如此重复下去。

练4.4　求阶乘n!的值

【题目描述】

利用for循环求n!的值。

提示，n!=1×2×...×n。

【输入格式】

输入一个正整数n。

【输出格式】

输出n!的值。

【输入样例】

4

【输出样例】

24

【题目分析】

分析n!＝1*2*3…*n，i和s初始值为1，然后把i乘到s中，i每次增量为1，一共把i乘到s中有n次。

【代码实现】

```
n=int(input()) # 输入给定1个整数n
s=1
for i in range(1,n+1):
    s*=i
print(s)
```

练4.5　数据统计

【题目描述】

输入一些整数，求出它们的最小值、最大值和平均值（保留3位小数）。输入保证这些数都是不超过1000的整数。

【输入格式】

一行，若干个整数。

【输出格式】

一行，即，最小值、最大值和平均值（保留3位小数）。

【输入样例】

```
1 2 3
```

【输出样例】

```
1 3 2.000
```

【代码实现】

```
# array是一个列表，里面按顺序存储了读入的数
array=input().split()

# 初始值设定
min=100000000 # 初始化最小值，赋比较大的数值
max=-100000000 # 初始化最大值，赋比较小的数值
n=0
s=0

for x in array:
    x=int(x)
    s+=x       # 每个数累加到s值中
    if x<min: # 如果x比min值小，就把x值保存到min中
        min=x
    if x>max: # 如果x比max值大，就把x值保存到max中
        max=x
    n+=1
print(min,max,"%.3f"%(s/n))
```

练4.6 三角形

【题目描述】

对于给定的自然数n(n<20)，在屏幕上输出仅由“*”构成的n行的直角三角形。例如：当n=5时，输出：

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

【输入格式】

输入n。

【输出格式】

题述三角形。

【输入样例】

5

【输出样例】

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

【题目分析】

打印图形总是逐行进行的，本题要重复n行操作，对于每一行，又重复若干次输出“*”操作。于是，构成了一个两层循环：外层循环是1至n行的处理，而内层循环，则是输出同一行上的每一列。分析样例，不难发现，每一行上“*”的个数恰好是行数。因此对于第i行，内层循环可以设置重复i次。

【代码实现1】

```
n=int(input())
for i in range(1,n+1):  # 外层循环，输出n行
    for j in range(i):  # 内层循环，处理每一行的输出
        print("*", end="")
    print("")
```

内层循环可以这样简化

【代码实现2】

```
n=int(input())
for i in range(1,n+1):
    print("*"*i)          # "*"*i表示"*"重复i次
```

第二节　while循环

当需要大量重复某段语句块而又不能确定重复次数的时候，我们可以利用while循环来编写程序，解决问题。其特点是：先判断条件表达式，后执行语句。

一、语句格式

```
while <条件表达式>:
    <循环体>
```

【说明】

(1)while表达式后面的冒号“:”不能省略。

(2)<条件表达式>一般是关系表达式或逻辑表达式，条件的值是逻辑真(True)或假(False)。

(3)如果条件为真，执行一次循环体，再次判断条件是否为真，如果仍为真，再执行一次循环体，以此类推，直到条件为假时退出while语句，执行循环体外的下一条语句(即while后没有缩进的第一条语句)。

(4)循环体中必须有改变<条件表达式>值的语句，否则循环体就会一直执行(死循环)。

二、例题应用

例4.6　募集捐款粤教

【题目描述】

某场募捐活动上，第一个人募捐20元，第二个人募捐25元……后一个人比前一个人均多募捐5元，求第几个人募捐后，总金额累计大等于500元。

【输入格式】

（无）

【输出格式】

一行一个整数，表示答案。

【代码实现】

```
num=1          # 统计捐款人数
money=20       # 第1个人捐款金额
sum=20         # 捐款金额
# 循环判断条件总金额累计小于500
while sum<500:
    num=num+1        # 捐款人数加1
    money=money+5    # 捐款金额比上一人加5元
    sum=sum+money    # 捐款金额总数
# 拉赞助人的数量为num
print(num)
```

【题目分析】

设拉赞助人的数量为num（num=1，2，3，…），每次募集的费用为money，每次募集之后总费用为sum，while循环枚举并累加每个人募捐金额，sum=20+25+30+…，当总金额大等于500时终止即可。

例4.7　数字反转沪版

【题目描述】

给定一个正整数，请将该数各位上数字反转得到一个新数。得到的新数的最高位数字不应为零，例如输入380，反转后得到的新数为83。

【输入格式】

一行一个正整数。

【输出格式】

一行一个正整数，表示反转后的新数。

【样例输入】

726

【样例输出】

627

【题目分析】

每次将原数对10取模可求出末位数，并除10取整消除末位数。将新数乘10+此数，即可将其添加至新数末尾，反复执行至原数为0即可。

【代码实现】

```
num=int(input())
# reverseNum是逆序数
reverseNum=0

while num>0:
# 取出num的最后一位，放在reverseNum的末尾
    reverseNum=reverseNum*10+num%10
    num=num//10
    # 注意要//表示整除
print(reverseNum)
```

例4.8　理财产品收益科教

【题目描述】

小明是一位精明的投资者。他购买了10万元一年期收益率3.7% 的银行保证收益型理财产品。每年理财赎回后，他会提取2万元用作生活所需，余下的自己仍购买此种理财，在收益率不变的情况下，多少年后被自己全部取出？

【输入格式】

(无)

【输出格式】

一行一个整数

【题目分析】

重复执行

money=round(money*(1+0.037),2)-20000

直到 money < 0 的时候终止循环，记下循环次数。

【代码实现】

```
money = 100000      # 原始理财资金10万
year = 0
while money >= 0:   # 还没有取完，继续理财
    money=round(money* (1+0.037), 2) -20000
    year += 1       # 理财年份加1
print(year)         # 输出理财年份
```

例4.9　求所有因子浙版

【题目描述】

输入一个正整数x，输出所有不包括本身的所有因子。

【输入格式】

输入一行一个整数x。

【输出格式】

输出若干行，一行一个整数，表示正整数x的不包括本身的所有因子。

【样例输入】

243

【样例输出】

1

3

9

27

81

【题目分析】

x的所有因数对x取模都为0，我们只要排除x本身即可。

【代码实现】

```
x=int(input()) # 输入给定1个正整数x

i=1
while i<x:
    # 如果i能整除x
    if x%i==0:
        print(i)
    i=i+1
```

例4.10　猜数游戏浙版

【题目描述】

请设计一个猜数游戏平台，有两个玩家Alice和Bob。Alice给定一个数，请Bob猜这个数是多少。猜数方法是交互的：Bob输入猜测的数字，平台给出相应的提示：“偏大”“偏小”或“正确”。若所猜数字正确，则游戏结束；否则继续猜数。

请你实现这个猜数游戏平台。

【输入格式】

第一行输入一个数，表示Alice给定的数。

接着输入若干行，表示Bob的猜数过程。每行一个整数，表示Bob猜测的数字。

【输出格式】

对于每一行输入，输出一行。输出偏大”“偏小”或“正确” 。

【样例输入】	【样例输出】
23	
300	偏大
20	偏小
100	偏大
25	偏大
23	正确

【题目分析】

按照题意进行模拟即可。

【代码实现】

```
# Alice给定数字number
number=int(input())
running=False

while not running:
    # Bob猜测数字guess
    guess=int(input())
    if guess==number:
        print("正确")
        running=True
    elif guess<number:
        print("偏小")
    else:
        print("偏大")
```

例4.11　计算圆周率沪版

【题目描述】

使用如下公式求圆周率π。

公式：$\frac{\pi^2}{6}=1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\cdots$

【输入格式】

无

【输出格式】

一行一个浮点数，表示圆周率π。

精确到小数点后4位。

【样例输入】

无

【样例输出】

3.1415

【题目分析】

套用题中公式，当右式枚举至累加项足够小时即可约等于左式。

【代码实现】

```
from math import sqrt
i=1
s=0
item=1
while item>0.00000001:
    # item 精确到0.1的8次方
    s=s+item
    i=i+1
    item=1/(i*i)
pi=sqrt(6*s)
# 精确到小数点后4位
print('%.4f'%(pi))
```

【课堂练习】

练4.7　第几项

【题目描述】

对于正整数n,m，求s=1+2+3……+n，当加到第几项时，s的值会超过m?

【输入】

输入m。

【输出】

输出n。

【输入样例】

1000

【输出样例】

45

【代码实现】

```
# 输入给定1个整数m
m=int(input())
n=0                 # 累加数n初始值为0
s=0                 # 累加和s初始值为0
while s<=m:         # 累加和s没超过m就继续累加
    n+=1            # 累加数n加1
    s+=n            # n累加到s中
print(n)            # 输出n的值
```

练4.8　求最大公约数

【题目描述】

求两个正整数m，n的最大公约数。

【输入格式】

输入m，n。

【输出格式】

m，n的最大公约数。

【输入样例】

4 6

【输出样例】

2

【题目分析1】

求任意两个自然数m和n的最大公约数，可以想到其最大的可能就是两个数中的较小者min，最小可能是1。所以，可以设最大公约数gcd从min开始进行判断，若gcd>1并且没有同时整除m和n，那么就gcd-1，重复判断是否整除。

【代码实现1】

```
# 当输入格式是一行两个整数时，用split()分割两个数字
m,n=input().split()
m=int(m)    # int(m)将字符串m转化为数字
n=int(n)    # int(n)将字符串n转化为数字
if m>n:     # 把m和n中值小的赋给gcd
    gcd=n
else:
    gcd=m
while gcd>1 and (m%gcd!=0 or n%gcd!=0):
    gcd-=1  # 每次减1寻找最大公约数
print(gcd)
```

练4.8　求最大公约数

【题目描述】

求两个正整数m，n的最大公约数。

【输入格式】

输入m，n。

【输出格式】

m，n的最大公约数。

【输入样例】

4 6

【输出样例】

2

【题目分析2】

求两个整数的最大公约数可以采用辗转相除法即欧几里德算法。对于任意两个自然数m和n，用m，n，r分别表示被除数、除数、余数，那么m和n的最大公约数等于n和r的最大公约数。以下是辗转相除法的算法：

(1)求m除以n的余数r。

(2)当r!=0,执行第(3)步；若r==0，则n为最大公约数，算法结束。

(3)将n的值赋给m，将r的值赋给n；再求m除以n的余数r。

(4)转到第(2)步。

【代码实现2】

```
# 当输入格式是一行两个整数时，用split()分割两个数字
m,n=input().split()
m=int(m)        # int(m)将字符串m转化为数字
n=int(n)        # int(n)将字符串n转化为数字
r=m%n
# 辗转相除法
while r!=0:
    m=n
    n=r
    r=m%n
print(n)
```

练4.9　求最小n值

【题目描述】

编一程序求满足不等式

$1 + \frac{1}{2} + \frac{1}{3} \ldots + \frac{1}{n}$ >= m的最小n值。

其中，n,m为正整数。

【输入格式】

输入m。

【输出格式】

输出n。

【输入样例】

3

【输出样例】

11

【题目分析】

此题不等式的左边是一个求和的算式，该和式中的数据项个数是未知的，也正是要求出的。对于和式中的每个数据项，对应的通式为$\frac{1}{i}$，i=1，2，…n。

所以可采用循环累加的方法来计算出它的值。设循环变量为i，它应从1开始取值，每次增加1，直到和式的值不小于m为止，此时的i值就是所求的n。设累加变量为s，在循环体内把$\frac{1}{i}$ 的值累加到s上。

【代码实现】

```
m=int(input()) # 输入给定1个整数m
i=0
s=0
# 当s的值还未超过m时
while s<m:
    i+=1
    s+=1/i
print(i)
```

练4.10　求末两位数

【题目描述】

求n个1992的乘积的末两位数是多少？

【输入格式】

输入n。

【输出格式】

如题述的末两位数。

【输入样例】

3

【输出样例】

88

【题目分析】

积的个位与十位数只与被乘数与乘数的个位与十位数字有关，所以本题相当于求n个92相乘，而且本次的乘积是下一次相乘的被乘数，因此只需取末两位参与运算就可以。

【代码实现】

```
# 输入给定1个整数m
n=int(input())
a=1
t=0
while t!=n:          # 还没乘n次就续继
    t+=1             # 统计乘法次数加1
    a=a*92%100       # 每次乘92的值取模后保留末两位数
print(a)             # 输出最后的结果
```

练4.11　体操队

【题目描述】

校体操队到操场集合，排成每行2人，最后多出1人；排成每行3人，也多出1人；分别按每行排4，5，6人，都多出1人；当排成每行7人时，正好不多。求校体操队至少多少人？

【输入格式】

如题述，无。

【输出格式】

校体操队人数。

例如：人数为15，直接输出15就可以啦。

【题目分析】

①设校体操队为x人，根据题意x应是7的倍数，因此x的初值为7，以后用x+=7改变x值；

②为了控制循环，用逻辑变量yes为真(true) 使循环结束；

③如果诸条件中有一个不满足，yes 的值就会为假(false)，就继续循环。

程序中对每个x值，都先给yes 赋真值，只有在循环体各句对x进行判断时，都得到“通过”（此处不赋假值）才能保持真值。

【代码实现】

```
x=0
yes=False
while yes==False:
    yes=True
    x+=7
    if x%2!=1:
        yes=False
    if x%3!=1:
        yes=False
    if x%4!=1:
        yes=False
    if x%5!=1:
        yes=False
    if x%6!=1:
        yes=False
print(x)
```

第三节　循环嵌套

一个循环结构可以包含另一个循环，这样的结构称为循环嵌套，也称多重循环。常用的循环嵌套是二重循环，外层循环称为外循环，内层循环称为内循环。内循环是外循环的循环体。循环嵌套的执行过程是要首先执行外层循环，外循环每执行一次，内循环则需执行一个完整的循环。例如，输出显示“九九乘法表”。

一、例题应用

例4.12　求阶乘和

【题目描述】

求 S=1!+2!+3!+....+n!

【输入格式】

输入正整数n。

【输出格式】

输出s。

【输入样例】

3

【输出样例】

9

【题目分析】

问题是求n以内自然数的阶乘之和，可以用for循环来实现。程序结构如下：

```
for i in range(1,n+1)
    (1)i阶乘的值存到t;  # 求t=i!
    (2)累加t到s中;      # s累加i!
```

根据以上结构，通过n次的循环可以求出1!，2!，…n!，并不断累加起来，求出s。而求t=i!,又可以用一个for循环来实现：

```
t=1
for j in range(1,i+1):    #求i!
    t=t*j
```

【代码实现】

```
n=int(input())
s=0
for i in range(1,n+1):
    t=1
    for j in range(1,i+1):  #求i!
        t=t*j
    s+=t    #累加i!到s中
print(s)
```

以上程序是一个for循环的嵌套。这种方法是比较容易想到的，但实际上对于求i!，我们可以根据求出的(i-1)!乘上i即可得到，而无需重新从1再累乘到i。

因此程序可改为:

```
n=int(input())
s=0
t=1
for i in range(1,n+1):
    # t为上一个数的i-1的阶乘值，再乘以i即为i!
    t*=i
    # 累加i!到s中
    s+=t
print(s)
```

显然第2个程序的效率要比第1个高得多。如果n=10，第1个程序要进行1+2+3+…+10=55次循环，而第2程序进行10次循环。若题目中求的是1!+2!+…+1000!，则两个程序的效率区别更明显。

例4.13　九九乘法表人教

参照以下输出格式

```
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
```

【代码实现】

```
# 外循环
for i in range(1,10):
    # 内循环
    for j in range(1,i+1):
        # 通过end = ""设置函数print()不输出换行符
        print(i,"*",j,"=",i*j," ",end = "")
    # 设置换行操作(函数print()不输出任何字符，只进行换行)
    print()
```

二、break语句和continue语句

在循环体中，可以使用break和continue跳转语句改变程序的执行顺序。

break语句的作用是强制退出循环，不再执行循环体内的语句，使程序跳到该循环结构之外的第一个可执行语句.

而continue语句的作用是结束本次循环，即跳过循环体内还未执行的语句，接着进行循环条件的判断，以决定是否继续执行循环。

例4.14　四位完全平方数

【题目描述】

输出所有形如aabb的四位完全平方数（即前两位数字相等，后两位数字也相等）。

【输入格式】

无

【输出格式】

由小到大输出，每个数占一行。

【题目分析】

分支和循环结合在一起时威力特别强大：我们枚举所有可能的aabb，然后判断它们是否为完全平方数。注意，a的范围是1～9，b可以是0。主程序如下：

```
for a in range(1,10):          # a的值
    for b in range(10):        # b的值
        if (aabb是完全平方数):
            print(aabb);
```

另一个思路是枚举平方根x，参考程序如下：

【代码实现】

```
x=1
while True:
    n=x*x
    x=x+1
    if n<1000:
        continue
    if n>9999:
        break
    high=n//100     # 注意使用整除符号//
    low=n%100
    if high//10==high%10 and low//10==low%10:
        print(n)
```

此程序中的新东西是continue和break语句。continue是指跳回while循环的开始，执行调整语句并判断循环条件，就是“直接进行下一次循环”，而break是指直接跳出循环。

另外，注意到这里的while语句是“残缺”的：没有指定循环条件。while True是一个死循环，如果不采取措施(如break)，它就永远不会结束。

【课堂练习】

练4.12　百钱买百鸡

【题目描述】

百钱买百鸡问题。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？

【输入格式】

无

【输出格式】

输出各种鸡翁、鸡母、鸡雏的数量，依次由小到大，每种情况各占一行，每行三个数之间用一个空格隔开。

练4.12　百钱买百鸡

【题目分析1】

在数学中解决这个问题，我们通常会列出一个方程组，设鸡翁x，鸡母y，鸡雏z，则：

x+y+z=100

5*x+3*y+z/3=100

同时满足上述两个方程的x、y、z值就是所求。

根据这个思路，问题就转化为求解方程组，我们列举x、y、z的所有可能解，然后判断这些可能解是否能使方程组成立。能使方程组成立的，就是真正的解。

再进一步分析，x的取值范围是1～100//5，y的取值范围是1～100//3，z的取值范围是1～3*100。

【代码实现1】

```
# 列举鸡翁数的所有可能
for x in range(100//5+1):
    # 列举鸡母数的所有可能
    for y in range(100//3+1):
        # 列举鸡雏数的所有可能，鸡雏的数量只能是3的整数倍
        for z in range(0,100*3+1,3):
            # 满足两个方程组
            if 5*x+3*y+z/3==100 and x+y+z==100:
                print(x,y,z)
```

运行结果：

```
0 25 75
4 18 78
8 11 81
12 4 84
```

练4.12　百钱买百鸡

【代码实现1】

```
# 列举鸡翁数的所有可能
for x in range(100//5+1):
    # 列举鸡母数的所有可能
    for y in range(100//3+1):
        # 列举鸡雏数的所有可能，鸡雏的数量只能是3的整数倍
        for z in range(0,100*3+1,3):
            # 满足两个方程组
            if 5*x+3*y+z/3==100 and x+y+z==100:
                print(x,y,z)
```

运行结果：

```
0 25 75
4 18 78
8 11 81
12 4 84
```

【说明】

这里用了一个三层循环的程序解决问题。当x取得一个数值时，for的y循环体都要执行遍y的所有取值；当y取得一个数值时，for的z循环体都要执行遍z的所有取值；对于z的每一个取值，if语句都要执行一次。

不难算出，在程序的执行过程中，作为最内层循环体的if语句，将被执行：(1+100/5)*(1+100/3)*(1+3*100)=214914次。而观察程序的运行结果，问题的解远远小于这个数字，只有4组解。如何减少循环次数呢？

练4.12　百钱买百鸡

【题目分析2】

由于题目的特殊性，鸡翁、鸡母、鸡雏共100只，一旦确定鸡翁x和鸡母y的数量，鸡雏便只能购买100-x-y只。这样，我们可以尝试写出一个两层循环的程序，解决这个问题。

【代码实现2】

```
# 列举鸡翁数的所有可能
for x in range(100//5+1):
    # 列举鸡母数的所有可能
    for y in range(100//3+1):
        # 根据x，y计算鸡雏的数量
        z=100-x-y
        # 满足两个方程组
        if 5*x+3*y+z/3==100 and z%3==0:
            print(x,y,z)
```

【说明】

对于与本题类似的求解不定方程问题，都可以用循环来求解。为提高效率，可以在程序中进行适当优化，减少循环体的执行次数。

练4.13　水仙花数

【题目描述】

求100－999中的水仙花数。若三位数ABC，$ABC=A^3+B^3+C^3$，则称ABC为水仙花数。

例如153，$1^3+5^3+3^3=1+125+27=153$，则153是水仙花数。

【输入格式】

无

【输出格式】

由小到大输出满足条件的数，每个数占一行。假设需要输出两个数：119,100。需要输出以下形式：

100

119

【题目分析】

根据题意，采用三重循环来求解。由于循环次数一定，用for循环最为简单。

【代码实现1】

```
for a in range(1,10):          # m的百位
    for b in range(10):        # m的十位
        for c in range(10):    # m的个位
            if a**3+b**3+c**3==a*100+b*10+c:
                print(a*100+b*10+c)
```

练4.13　水仙花数

同时也可以采用一个for循环来求解，表面上看好像优于三重循环，实际上却比上面的程序效率低，请同学们自己分析。

【代码实现2】

```
for m in range(100,1000):
    # //表示整除
    # m的百位
    a=m//100
    # m的十位
    b=(m%100)//10
    # m的个位
    c=m%10
    # a**3表示a的三次方
    if (a**3+b**3+c**3==m):
        print(m)
```

练4.14　找素数

【题目描述】

输出正整数a到b之间的所有素数。

【输入格式】

输入a,b。

【输出格式】

由小到大，输出a到b之间的所有素数。每个数占一行。

【输入样例】

5 10

【输出样例】

5

7

【题目分析】

对100-200之间的每一个整数进行判断，若它是为素数，则输出。而对于任意整数i，根据素数定义，从2开始，到sqrt（i），找i的第一个约数，若找到第一个约数，则i必然不是素数。

【代码实现】

```
a,b=map(int,input().split())  #输入a和b的值
from math import sqrt
for i in range(a,b+1):
    x=2
    # 在枚举的范围内并且没有出现约数则继续枚举
    while x<=int(sqrt(i)) and i%x!=0:
        x+=1
    if x>int(sqrt(i)):
        print(i)
```

练4.15　阶乘之和

【题目描述】

输入n，计算S=1! + 2! + 3! + … + n!的末6位(不含前导0)。n<=10^6，n!表示前n个正整数之积。

【输入格式】

输入n。

【输出格式】

如题述，求阶乘之和。

【输入样例】

10

【输出样例】

37913

【题目分析】

引入累加变量s之后，核心算法只有一句话：

```
for i in range(1,n+1):
    s+=i!。
```

还需要一次循环来计算i!：

```
for j in range(1,i+1):
    fac*=j。
```

【代码实现】

```
n=int(input())
s=0
for i in range(1,n+1):
    fac=1
    for j in range(1,i+1):
        fac*=j    # fac是i!
    s+=fac
print(s%1000000)
```

注意累乘器factorial(英文“阶乘”的意思)定义在循环里面。换句话说，每执行一次循环体，都要重新声明一次factorial，并初始化为1(想一想，为什么不是0)。因为只要末6位，所以输出时对10^6取模。

练4.16　分解质因数

【题目描述】

把一个合数分解成若干个质因数乘积的形式(即求质因数的过程)叫做**分解质因数**。分解质因数(也称分解素因数)只针对合数。

输入一个正整数n，将n分解成质因数乘积的形式。

【输入格式】

一个正整数n。

【输出格式】

分解成质因数乘积的形式。质因数必须由小到大，见样例。

【输入样例】

36

【输出样例】

36=2*2*3*3

【题目分析】

将任意的n分解为质因数的乘积，要从最小的质数开始，那么，我们就不妨从2开始试除，能整除就输出2，再对商继续试除,直到不再含有因子2；然后用下一个质数反复试除，……，再用下一个质数试除，……，一直到商为1，停止操作。

这里，质因数的递增，是一层循环，每一个质因数的反复试除，又是一层循环。因此，本题使用两层循环来解决。

【代码实现】

```
n=int(input())
i=2
print(n,end="=")
while n!=1:       # n没有除尽，就重复操作
    while n%i==0:# n能被i整除，就重复做除法操作
        print(i,end="")
        n/=i
        if n!=1:
            print("*",end="")
    i+=1
```

练4.17　最佳购买方案粤教

【题目描述】

某小组欲以1000元购买1.8元，1.9元，2.1元的纪念品若干件，每种记录品至少买100件，设计一个方案，使购买的纪念品最多，若有多种方案，选择余额最小的方案。

【输入格式】

（无）

【输出格式】

每行分别表示方案中每一种物品的数量，总数量和所剩余额。

【样例输入】

（无）

【样例输出】

（无）

【题目分析】

循环枚举每种购买数量，并用if语句判断新的方案是否合法，更优。

练4.17　最佳购买方案(粤教)

【代码实现】

```
x1=y1=z1=100
s=300
r=1000-(100*1.8+100*1.8+100*2.1)
for x in range(100,556):
    for y in range(100,527):
        for z in range(100,277):
            if 1.8*x+1.9*y+2.1*z<=1000:
                if x+y+z>s:
                    s=x+y+z
                    r=1000-(1.8*x+1.9*y+2.1*z)
                    x1=x
                    y1=y
                    z1=z
                if x+y+z==s and r>=1000-(1.8*x+1.9*y+2.1*z):
                    s=x+y+z
                    r=1000-(1.8*x+1.9*y+2.1*z)
                    x1=x
                    y1=y
                    z1=z
            else:
                break    # 当1.8*x+1.9*y+2.1*z>1000时可以退出循环，减少时间浪费
print(x1) #单价为1.8元的物品数量
print(y1) #单价为1.9元的物品数量
print(z1) #单价为2.1元的物品数量
print(s)  #买到礼物的总数量
print(r)  #余款
```