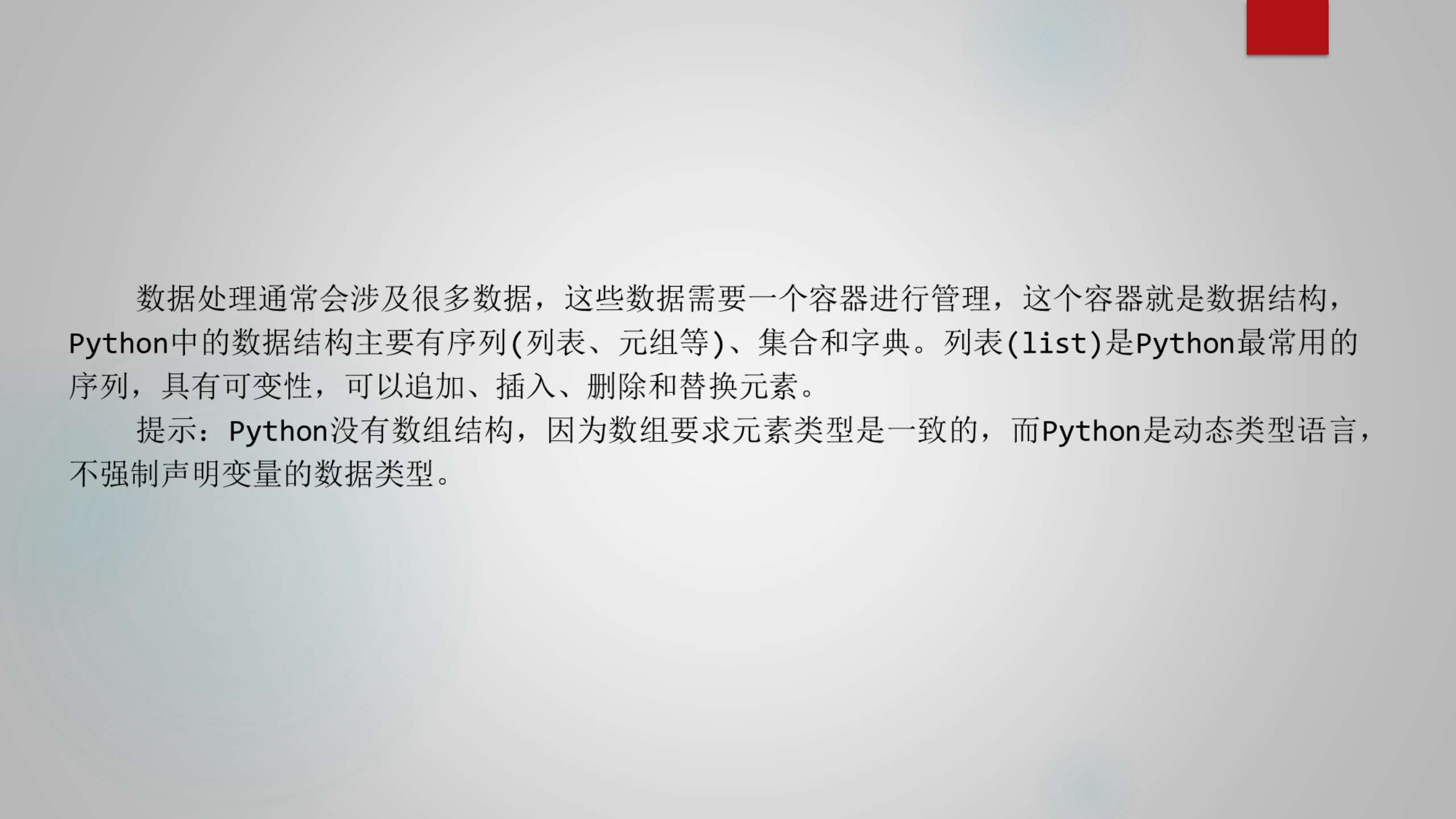


# Python一本通

信息技术学考教程

# 第5章 列表 元组 字典



数据处理通常会涉及很多数据，这些数据需要一个容器进行管理，这个容器就是数据结构，Python中的数据结构主要有序列(列表、元组等)、集合和字典。列表(`list`)是Python最常用的序列，具有可变性，可以追加、插入、删除和替换元素。

提示：Python没有数组结构，因为数组要求元素类型是一致的，而Python是动态类型语言，不强制声明变量的数据类型。

# 一、列表

列表是一组有序存储的数据。创建列表使用方括号“[]”将元素括起来，元素之间用逗号分隔，定义列表格式如下：

列表名称 = [元素1,元素2,元素3,...]

提示：创建空列表用“[]”表示。

## 【说明】

- (1)列表中的元素可以是任意数据类型。
- (2)列表的索引、切片等操作与字符串相同(详见第七章字符串索引与切片)
- (3)列表是可变的，即可以删除、添加或修改列表元素。
- (4)可以使用list()函数将字符串或元组转换为列表。
- (5)列表是一种有序的数据类型，其下标编号可以从左到右编号，也可以从右到左编号。

例如:

```
fruit=["葡萄","桃子","西瓜","橙子"]      #定义列表并初始化
```

下标从左到右编号: 0      1      2      3

下标从右到左编号: -4     -3     -2     -1

如果要将列表中的某个元素取出来, 使用“列表名[下标]”取出来, 如:

```
>>>fruit[0]      #访问列表中的第1个元素
```

```
'葡萄'
```

```
>>> fruit[0] = "苹果"      #修改列表第1个元素, 用"苹果"替换"葡萄"
```

```
>>>fruit
```

```
['苹果', '桃子', '西瓜', '橙子']      #显示更新后的列表
```

```
>>>fruit[::2]
```

```
['苹果', '西瓜']      #切片操作, 返回从左→右, 步长为2的元素
```

```
>>>del fruit[1]      #使用del语句删除列表中第2个(下标为1)元素
```

```
>>>fruit
```

```
['苹果', '西瓜', '橙子']
```

```
>>>list("abcd")
```

```
['a', 'b', 'c', 'd']
```

## 二、元组

元组与列表一样，也是一组有序存储的数据。创建元组使用小括号“()”将元素括起来，元素之间用逗号分隔，定义元组格式如下：

元组名称 = (元素1, 元素2, 元素3, ...)

### 【说明】

- (1)与列表类似，元组具有列表的大多数特点，如元组中的元素可以是任意数据类型，元组的索引、切片等操作与列表相同。
- (2)与列表不同的是，元组元素是不可变的，不能删除、添加或修改列表元素。
- (3)可以使用`tuple()`函数将列表转换为元组。

例如:

```
>>>fruit = ( "葡萄","桃子","西瓜","橙子" )      #定义元组并初始化
```

```
>>>fruit[0]      #访问元组中的第1个元素
```

```
'葡萄'
```

```
>>>fruit[1:3]
```

```
('桃子', '西瓜')      #切片操作, 返回索引下标从1→3(不包含3)
```

```
>>> len(fruit)      #计算元组的长度, 即元组元素个数
```

```
4
```

```
>>> tuple(fruit)      #使用tuple()函数将列表转换元组
```

```
('葡萄', '桃子', '西瓜', '橙子')
```

## 三、字典

映射(在数学里)是指两个元素的集合之间,元素相互“对应”的关系。

字典就是一种无序的映射集合。字典由一系列“键:值”对组成,键值之间用冒号分隔开。字典的“键值对”必须放大括号中,每组“键值对”以逗号隔开,定义列表格式如下:

字典名称={键1:值1,键2:值2,键3:值3,...}

### 【说明】

- (1)字典是一个无序的“键:值”对集合。其中“键”必须是不可变的数据类型,而“值”可以是任意数据类型。
- (2)在同一个字典中,“键”必须是唯一的,而“值”是可以重复的。
- (3)因字典无序的特性,故不能像字符串、列表、元组等一样使用索引和切片操作。字典通过“键”来获取“值”。
- (4)字典长度是可变的,即可以进行添加或删除“键:值”对。
- (5)可以使用dict()函数定义字典。



例如:

```
>>>student={ "姓名":"李白" , "爱好":"喝酒" , "学号":1001 }
>>>student          #显示字典中所有“键:值”对集合
{'姓名': '李白', '爱好': '喝酒', '学号': 1001}

>>>student["姓名"]  #通过键“姓名”来索引值"李白"
'李白'

>>>student["姓名"] = "张三"  #修改值
>>>student
{'姓名': '张三', '爱好': '喝酒', '学号': 1001}

>>>del student["爱好"]      #使用del语句删除字典元素
>>>student
{'姓名': '张三', '学号': 1001}
```

## 四、for语句使用列表

### 【例5.1】输出hello和world<sup>科教</sup>

#### 【题目描述】

通过for语句输出列表中的每个元素。

#### 【输入格式】

(无)

#### 【输出格式】

输出列表中的每个元素。

#### 【样例输入】

(无)

#### 【样例输出】

hello

world

#### 【题目分析】

循环体只有一条语句，即`print(i)`。循环执行时，依次将列表中的元素赋给`i`，并且输出`i`的值，直到整个列表的元素补取完，循环结束。

#### 【代码实现】

```
for i in ["hello","world"]:  
    print(i)                #循环体
```

运行结果：

hello

world

## 四、for语句使用列表

### 【例5.2】输出姓名<sup>人教</sup>

#### 【题目描述】

在for语句直接使用列表等序列，输出列表中的姓名。

#### 【输入格式】

(无)

#### 【输出格式】

输出列表中的姓名。

#### 【样例输入】

(无)

#### 【样例输出】

张明  
王丽  
李刚  
唐华

#### 【题目分析】

首先创建列表names，并给列表赋值。循环体只有一条语句，即print(name)。循环执行时，依次将列表中的元素赋给name，并且输出name的值，直到整个列表的元素补取完，循环结束。

#### 【代码实现】

```
names = ["张明", "王丽", "李刚", "唐华"]  
for name in names:  
    print(name)           #循环体
```

运行结果:

张明  
王丽  
李刚  
唐华

# 五、例题深入应用

## 【例5.3】反序输出

### 【题目描述】

输入n个数,要求程序按输入时的逆序把这n个数打印出来,已知整数不超过100个。也就是说,按输入相反顺序打印这n个数。

### 【输入格式】

输入一行共有n个数,每个数之间用空格隔开。

### 【输出格式】

如题要求:一行,共有n个数,每个数之间用一个空格隔开。

### 【输入样例】

1 7 3 4 5

### 【输出样例】

5 4 3 7 1

### 【题目分析】

定义一个列表a用以存放输入的n个数,然后将列表a中的内容逆序输出。

### 【代码实现】

```
# 将输入分割为多个字符串,储存在字符串列表a中
a=input().split()
#从后往前枚举,枚举到位置为-1结束。相当于倒序枚举
for x in a[::-1]:
    print(x,end=' ')
```

## 【例5.4】5年储蓄存款收益<sup>科教</sup>

### 【题目描述】

小明是一位理财专家，由于他的股票连年亏损，他决定把他的钱存入银行，并且他选择“一年定期，自动转存”。首次存款的利率是3.25%，自动转存时的利率分别是3%，3%，2%，1.75%，请问五年后到期存款总额是多少？

### 【输入格式】

程序输入一个整数，表示小明的本金金额。

### 【输出格式】

输出五年后到期存款总额是多少。四舍五入保留两位小数。

### 【样例输入】

50000

### 【样例输出】

56841.97

### 【题目分析】

将本金和所有的利润加起来即可。

### 【代码实现】

```
rate=[0.0325,0.03,0.03,0.02,0.0175]
money=int(input())
for i in range(5):
    money*=(rate[i]+1)
    # 五年后到期存款总额
print("%.2f"%(money))
```

## 【例5.5】冒泡排序

### 【题目描述】

编程输入 $n$  ( $1 \leq n \leq 20$ ) 个小于1000非负整数，然后自动按从大到小的顺序输出。（冒泡排序）

### 【输入格式】

第一行，数的个数 $n$ 。

第二行， $n$ 个非负整数。

### 【输出格式】

由大到小的 $n$ 个非负整数，每个数占一行。

### 【输入样例】

5

2 5 8 6 12

### 【输出样例】

12

8

6

5

2

## 【例5.5】冒泡排序

### 【题目分析】

①把n个数输入到a列表中；

②从a[1]到a[n]，相邻的两个数两两相比较，即：

a[1]与a[2]比，a[2]与a[3]比，.....a[n-1]与a[n]比。

只需知道两个数中的前面那元素的标号，就能进行与后一个序号元素（相邻数）比较，可写成通用形式a[i]与a[i+1]比较，那么，比较的次数又可用 $1 \sim (n-i)$ 循环进行控制(即循环次数与两两相比较时前面那个元素序号有关)；

③在每次的比较中，若较大的数在后面，就把前后两个对换，把较大的数调到前面，否则不需调换位置。

## 【例5.5】冒泡排序

### 【题目分析】

下面例举5个数来说明两两相比较和交换位置的具体情形：

5 6 4 3 7

5和6比较，交换位置，排成下行的顺序；

6 5 4 3 7

5和4比较，不交换，维持同样的顺序；

6 5 4 3 7

4和3比较，不交换，顺序不变

6 5 4 3 7

3和7比较，交换位置，排成下行的顺序；

6 5 4 7 3

经过（1～（n-1））次比较后，将3调到了末尾。

经过第一轮的比较，就能把十个数中的最小数调到最末尾位置，第二轮比较1～（N-2）次进行同样处理，又把这一轮所比较的“最小数”调到所比较范围的“最末尾”位置；.....；每进行一轮两两比较后，其下一轮的比较范围就减少一个。最后一轮仅有一次比较。在比较过程中，每次都有一个“最小数”往下“掉”，用这种方法排列顺序，常被称之为“冒泡法”排序。



## 【例5.5】冒泡排序

### 【代码实现】

```
import math
n=int(input())
a=input().split()      # 将输入分割为多个字符串，储存在字符串列表a中
a=map(int,a)           # 将字符串列表a中的每个元素转化为整数
a=list(a)              # list()函数将a转化为列表

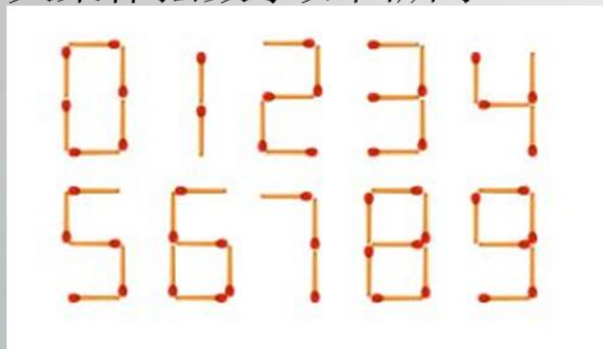
# 冒泡法排序
for i in range(1,n):
    for j in range(n-i):
        # 两两相比较
        if a[j]<a[j+1]:
            # 交换
            a[j],a[j+1]=a[j+1],a[j]
for i in a:
    print(i)
```

## 【例5.6】火柴数字问题<sup>科教</sup>

### 【题目描述】

你要从0到111输出所有恰好需要6根火柴棒摆出的所有数。

火柴棒摆数字如图所示：



### 【输入格式】

(无)

### 【输出格式】

输出若干行，每行一个数，表示0到111输出所有恰好需要6根火柴棒摆出的所有数。

### 【题目分析】

先建立一个函数求出某个数所需要的火柴棒个数。

### 【代码实现】

```
cnt = [6, 2, 5, 5, 4, 5, 6, 3, 7, 6]
snum = 6          # 6 根火柴棒
for num in range(112):
    #计算数字num需要多少火柴棒，记作bnum
    bnum=0
    if num<10:
        bnum= cnt[num]
    elif num<100:
        bnum=cnt[num//10] + cnt[num%10]
    else:
        bnum=cnt[num//100] + cnt[num//10 % 10]
        + cnt[num % 10]
    #如果i需要的火柴棒数与现有火柴棒数相等
    if bnum == snum:
        print(num)
```

# 【课堂练习】

## 练5.1 输出兴趣爱好<sup>浙版</sup>

### 【题目描述】

在for语句直接使用列表等序列，输出列表中的兴趣爱好。

### 【输入格式】

(无)

### 【输出格式】

输出列表中的兴趣爱好。

### 【样例输入】

(无)

### 【样例输出】

篮球  
羽毛球  
看书  
旅游  
音乐

### 【题目分析】

循环体只有一条语句，即print(x)。循环执行时，依次将列表中的元素赋给x，并且输出x的值，直到整个列表的元素补取完，循环结束。

### 【代码实现】

```
hobby = ["篮球", "羽毛球", "看书", "旅游", "音乐"]  
for x in hobby:  
    print(x)           #循环体
```

## 练5.2 平移数据

### 【题目描述】

将列表a中第一个元素移到列表末尾，其余数据依次往前平移一个位置。

### 【输入格式】

第一行为列表a的元素个数；

第二行为n个小于1000的正整数。

### 【输出格式】

平移后的列表元素，每个数用一个空格隔开。

### 【输入样例】

```
10
1 2 3 4 5 6 7 8 9 10
```

### 【输出样例】

```
2 3 4 5 6 7 8 9 10 1
```

### 【题目分析】

为完成题目所要求的操作，其算法应该包括以下几个主要步骤：

①把第一个元素的值取出放在一个临时单元 temp中；

②通过  $a[1] \rightarrow a[0]$ ,  $a[2] \rightarrow a[1]$ ,  $a[3] \rightarrow a[2]$ , ……,  
 $a[n-1] \rightarrow a[n-2]$ , 实现其余元素前移。

③将 temp值送入a[n-1]。

### 【代码实现】

```
n=int(input())           #输入元素个数n
# 将输入分割为多个字符串，储存在字符串列表a中
a=input().split()
temp=a[0]                #第1个元素临时保存在temp
for i in range(n-1):
    a[i]=a[i+1]          #每个元素送入前一个元素位置
a[n-1]=temp              #把临时保存temp送入最末尾
for i in range(n):
    print(a[i],end=' ')
```

## 练5.3 开关房门

### 【题目描述】

宾馆里有 $n$  ( $2 \leq n \leq 1000$ ) 个房间，从1~ $n$ 编了号。第一个服务员把所有的房间门都打开了，第二个服务员把所有编号是2的倍数的房间“相反处理”，第三个服务员把所有编号是3的倍数的房间作“相反处理”...，以后每个服务员都是如此。当第 $n$ 个服务员来过后，哪几扇门是打开的。（所谓“相反处理”是：原来开着的门关上，原来关上的门打开。）

### 【输入格式】

房间数 $n$ 。

### 【输出格式】

一行，由小到大的打开门的房间序号，各序号之间用一个空格隔开。

### 【输入样例】

100

### 【输出样例】

1 4 9 16 25 36 49 64 81 100

### 【题目分析】

此题较简单，用 $a[1]$ ,  $a[2]$ , ...,  $a[n]$ 表示编号为1, 2, 3, ...,  $n$ 的门是否开着。模拟这些操作即可。

### 【代码实现】

```
n=int(input())          #输入个数n
# 设置一个长度为n+1的列表，元素的初值是False
a=[False]*(n+1)
for i in range(1,n+1):
    for j in range(1,n+1):
        # j是i的倍数，那么第j号门被“相反处理”
        if j%i==0:
            # 相反处理
            a[j]=not a[j]
for i in range(1,n+1):
    if a[i]:
        print(i,end=' ')
```

## 练5.4 最大数位置

### 【题目描述】

输入n个整数,存放在数组a[1]至a[n]中,输出最大数所在位置( $n \leq 1000$ )。

### 【输入格式】

第一行, 数的个数n;

第二行, n个正整数, 每个数在 $2^{32}-1$ 之内。

### 【输出格式】

最大数所在位置。

### 【输入样例】

5

67 43 90 78 32

### 【输出样例】

3

### 【题目分析】

设maxa存放最大值, k存放对应最大值所在的列表位置, maxa的初值为a[1], k的初值对应为1, 枚举列表元素, 找到比当前maxa大的数成为maxa的新值, k值为对应位置, 输出最后的k值。

### 【代码实现】

```
n=int(input())
# 将输入分割为多个字符串, 储存在字符串列表a中
a=input().split()
# 将字符串列表a中的每个元素转化为整数
a=list(map(int,a))
# list()函数将a转化为列表
a=list(a)
# 赋最大值初值和初始位置
maxa=a[0]
k=0
# 枚举列表, 找到最大数和位置
for i in range(n):
    if a[i]>maxa:
        maxa=a[i]
        k=i
# 最大数所在列表中的位置
print(k+1)
```

## 练5.5 运动消耗热量表<sup>浙版</sup>

### 【题目描述】

小明是一位学生，他很会运动。下面是他一天的运动消耗热量表：

运动项目	慢走	骑自行车	打羽毛球	爬楼梯	跳绳	慢跑
消耗热量 (单位：大卡)	95	100	122	180	245	221

你需要输出他运动消耗热量总和。

### 【输入格式】

(无)

### 【输出格式】

输出一行一个整数，表示他运动消耗热量总和。

### 【样例输入】

(无)

### 【样例输出】

963

### 【题目分析】

直接将热量总和相加即可。

### 【代码实现】

```
a=[95,100,122,180,245,221]
s=0
# 枚举i为列表a中的数
for i in a:
    s+=i
print(s)
```

## 练5.6 筛选法找质数

### 【题目描述】

用筛法求出 $n(2 \leq n \leq 1000)$ 以内的全部质数。

### 【输入格式】

输入 $n$ 。

### 【输出格式】

输出若干个数字，用空格隔开，表示由小到大的质数。

### 【输入样例】

10

### 【输出样例】

2 3 5 7

### 【题目分析】

质数筛法是这样的：

原理很简单，就是当 $i$ 是质数的时候， $i$ 的所有的倍数必然是合数。如果 $i$ 已经被判断不是质数了，那么继续找到 $i$ 后面的质数来把这个质数的倍数筛掉。

一个简单的筛素数的过程： $n=30$ 。

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21  
22 23 24 25 26 27 28 29 30

第 1 步过后 2 4 ... 28 30 这 15 个单元被标成 false, 其余为 true。

第 2 步开始：

$i=3$ ；由于  $\text{prime}[3]=\text{true}$ ，把  $\text{prime}[6]$ ,  $[9]$ ,  $[12]$ ,  $[15]$ ,  $[18]$ ,  $[21]$ ,  $[24]$ ,  $[27]$ ,  $[30]$  标为 false。

$i=4$ ；由于  $\text{prime}[4]=\text{false}$ ，不在继续筛法步骤。

$i=5$ ；由于  $\text{prime}[5]=\text{true}$ ， $\text{prime}[10]$ ,  $[15]$ ,  $[20]$ ,  $[25]$ ,  $[30]$  标为 false。

$i=6$ ； $>\text{sqrt}(30)$  算法结束。

第 3 步把  $\text{prime}[]$  值为 true 的下标输出来：

结果是 2 3 5 7 11 13 17 19 23 29



## 练5.6 筛选法找质数

### 【题目描述】

用筛法求出 $n$  ( $2 \leq n \leq 1000$ ) 以内的全部质数。

### 【输入格式】

输入 $n$ 。

### 【输出格式】

输出若干个数字，用空格隔开，表示由小到大的质数。

### 【输入样例】

10

### 【输出样例】

2 3 5 7

### 【代码实现】

```
import math
n=int(input())      #输入个数n
# 设置一个长度为n+1的列表，列表元素的初值是True
prime=[True]*(n+1)
prime[1]=False
maxn=int(math.sqrt(n)+1)
for i in range(2,maxn):
    if prime[i]:
        for j in range(2,n//i+1):
            # i*j是合数
            prime[i*j]=False
for i in range(2,n+1):
    if prime[i]:
        print(i,end=' ')
```

## 练5.7 约瑟夫问题

### 【题目描述】

N个人围成一圈，从第一个人开始报数，数到M的人出圈；再由下一个人开始报数，数到M的人出圈；... 输出依次出圈的人的编号。

### 【输入格式】

输入N和M。

### 【输出格式】

输出一行，依次出圈的人的编号。

### 【输入样例】

8 5

### 【输出样例】

5 2 8 7 1 4 6 3

### 【题目分析】

①由于对于每个人只有出圈和没有圈两种状态，因此可以用布尔型标志列表存储游戏过程中每个人的状态。不妨用true表示出圈，false表示没有出圈。

②开始的时候，给标志列表元素赋初值为false，即全部在圈内。

③模拟报数游戏的过程，直到所有的人出圈为止。

## 练5.7 约瑟夫问题

### 【代码实现】

```
n,m=input().split()      # 共n人，报到m出圈
n=int(n)
m=int(m)
a=[False]*(n+1)         # 设置一个列表a[0..n]，列表元素的初值是false
f,t,s=0,0,0
while f!=n:             # 直到所有的人都出圈为止
    t+=1                 # 逐个枚举圈中的所有位置
    if t==n+1:           # 列表模拟环状，最后一个与第一个相连
        t=1
    if a[t]==False:      # 第t个位置上有人则报数
        s+=1
    if s==m:             # 当前报的数是m
        s=0              # 计数器清零
        print(t,end=' ') # 输出出圈人的编号
        a[t]=True        # 此处的人已出圈，设置为空
        f+=1             # 出圈的人数增加一个
```