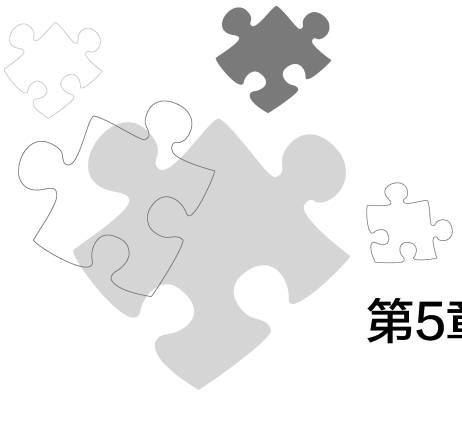

第5章　列表和元组

5.1　知 识 要 点

5.1.1　序列

Python 中常用的序列结构有列表、元组、字典、集合、字符串等。从是否有序这个角度看，Python 序列可以分为有序序列和无序序列，其中，列表、元组和字符串属于有序序列，而字典和集合属于无序序列；从是否可变来看，Python 序列则可以分为可变序列和不可变序列两大类。其中，列表、字典和集合属于可变序列，而元组和字符串属于不可变序列。列表、元组、字符串有序序列支持双向索引，第一个元素下标为 0，第二个元素下标为 1，以此类推；如果使用负数作为索引，则最后一个元素下标为－1，倒数第二个元素下标为－2，以此类推。

5.1.2　列表

1. 列表的创建和删除

1）列表的创建

列表的创建可以使用 list 类的构造函数创建一个空列表，也可以使用“＝”直接创建一个列表并赋值给其他变量，其格式如下。

```
格式 1: 列表名 = list()
格式 2: 列表名 = [元素 1, 元素 2, …, 元素 n]
```

2）列表的删除

当列表不用时可以使用 del 命令将其删除，其格式如下。

```
del 列表名
```

2. 列表元素的访问

在列表创建之后，就可以进行使用了。列表元素的使用可以使用下标法，其格式如下。

```
列表名[下标]
```

使用整数作为下标来访问列表的元素，下标从 0 开始，第一个元素是 a_list[0]，第 2 个元素是 a_list[1]，以此类推；列表还支持使用负整数作为下标，其中，下标为－1 的元素表示最后一个元素，下标为－2 的元素表示倒数第 2 个元素，以此类推。

3. 列表的基本操作

列表、元组、字典、集合、字符串等序列有很多操作是通用的，而不同类型的序列又有一些特有的方法或者支持某些特有的运算符和内置函数。列表常用的操作如表 5-1 所示。

表 5-1　列表常用的操作

方　法	说　明
append(x)	将 x 追加至列表尾部
extend(L)	将列表 L 中的所有元素追加至列表尾部
insert(index, x)	在列表 index 位置处插入 x
remove(x)	在列表中删除第一个值为 x 的元素
pop([index])	删除并返回列表中下标为 index 的元素，省略 index 时，删除最后一个元素
clear()	清空列表，删除列表中的所有元素，保留列表对象
index(x)	返回列表中第一个值为 x 的元素的索引
count(x)	返回 x 在列表中的出现次数
reversed( )	对列表所有元素进行原地逆序，首尾交换
sort(key＝None, reverse＝False)	对列表中的元素进行原地排序，key 用来指定排序规则，reverse 为 False 表示升序，为 True 表示降序
copy()	返回列表的浅复制

4. 列表对象支持的运算符

1）加法运算

加法运算符＋和复合加法赋值＋＝用于将两个列表合并。需要注意的是，“＋”运算不属于原地操作，而是返回一个新列表，效率比较低。“＋＝”运算属于原地操作，与 append() 方法一样高效。

2）乘法运算

乘法运算符 * 用于列表和整数相乘，表示序列重复，不属于原地操作，返回新列表。复合乘法赋值 * ＝也可用于列表元素的重复，与运算符＋＝一样也属于原地操作。

3）成员测试运算符 in

成员测试运算符 in 可用于测试列表中是否包含某个元素，结果是 bool 型。

5. 列表推导式

列表推导式的语法形式为：

```
[expression for expr1 in sequence1 if condition1
            for expr2 in sequence2 if condition2
            for expr3 in sequence3 if condition3
                          ⋮
            for exprN in sequenceN if condtionN]
```

列表推导式在逻辑上等价于一个循环语句，只是形式上更加简洁。

6. 列表的切片操作

切片操作除了适用于列表之外，还适用于元组、字符串。切片使用两个冒号分隔的 3 个数字来完成，其格式如下。

```
[start: end: step]
```

其中，第一个数字 start 表示切片开始的位置，默认为 0；第二个数字 end 表示切片截止（但不包含）的位置，默认为列表长度；第三个数字 step 表示切片的步长（默认为 1）。当 start 为 0 时可以省略，当 end 为列表长度时可以省略，当 step 为 1 时可以省略，省略步长时还可以同时省略最后一个冒号。另外，当 step 为负整数时，表示反向切片，这时 start 应该在 end 的右侧才行。

7. Python 内置函数对列表的操作

通过内置函数可以对列表进行操作。例如，len( )、max( )、min( )可以获取列表的长度、列表中元素的最大值和最小值；sum( )可以获取列表或元组中元素之和；zip( )函数用于将多个列表中元素重新组合为元组并返回包含这些元组的 zip 对象；map()函数把函数映射到列表上的每个元素等。

5.1.3 元组

1. 元组的创建、删除与元素访问

1）元组的创建

在形式上，元组的所有元素放在一对圆括号中，元素之间使用逗号分隔，如果元组中只有一个元素则必须在最后增加一个逗号。元组的创建格式如下。

```
格式 1: 元组名 = tuple()
格式 2: 元组名 = (元素 1, 元素 2, … , 元素 n)
```

2）元组的删除

与列表的删除一样，当不再使用元组时，可以使用 del 命令将其删除，其格式如下。

```
del 元组名
```

3）元组元素的访问

在元组创建之后，就可以进行使用了。元组元素的使用也是下标法，其格式如下。

```
元组名[下标]
```

元组也支持双向索引。元组属于不可变序列，不可以修改、添加、删除元素。

2. 元组与列表的异同点

1）相同点

元组和列表都属于有序序列，都支持使用双向索引访问其中的元素。

2）不同点

列表属于可变序列，而元组属于不可变序列。因此，元组没有 append( )、extend( )、

insert( )等方法，也没有 pop( )、remove( )方法，同时，元组也不支持对元素进行 del 操作，不能从元组中删除元素，而只能使用 del 删除整个元组。

元组也支持切片操作，但只能通过切片来访问元组中的元素，不能使用切片改变元组中的元素值。从一定程度上来说，元组是轻量级的列表。

Python 的内部实现对元组做了大量优化，访问速度比列表快。

5.2　例题分析与解答

一、选择题

1. 下列哪种不是 Python 元组的定义方式？_____

A. (1)　　B. (1,)　　C. (1, 2)　　D. (1, 2, (3, 4))

分析：Python 中规定，如果元组中只有一个元素，则必须在最后增加一个逗号，所以 A 选项的定义方式是错的。

答案：A

2. 下列代码执行结果是 _____。

```
[i ** i for i in range(3)]
```

A. [1, 1, 4]　　B. [0, 1, 4]　　C. [1, 2, 3]　　D. (1, 1, 4)

分析：** 是 Python 的幂运算符，i ** i 表示 i 的平方，[i ** i for i in range(3)]是列表推导式，i 的取值范围是 0～2，即对 0,1,2 分别求平方，需要注意的是 $0^2=1$，所以答案应该是 A 选项。

答案：A

3. Python 语句 print(type([1,2,3,4])的输出结果是_____。

A. <class 'tuple'>　　B. <class 'dict'>

C. <class 'set>　　D. <class 'list'>

分析：内置函数 type()用来查看变量的类型，很明显[1,2,3,4]是列表 list，所以选 D 选项。

答案：D

4. Python 语句 print(type((1,2,3,4))的输出结果是_____。

A. <class 'tuple'>　　B. <class 'dict'>

C. <class 'set>　　D. <class 'list'>

分析：内置函数 type()用来查看变量的类型，很明显(1,2,3,4)是元组 tuple，所以选 A 选项。

答案：A

5. Python 语句 print(type({}))的输出结果是_____。

A. <class 'tuple'>　　B. <class 'dict'>

C. <class 'set>　　D. <class 'list'>

分析：内置函数 type()用来查看变量的类型，花括号“{}”既可以是字典的定界符，也可以是集合的定界符，如果花括号内没有任何元素，则认为是一个空字典，所以选 B 选项。

答案：B

6. Python 语句 x=[1,2,3,None,(),[],]; print(len(x))的输出结果是_____。

A. 4　　B. 5　　C. 6　　D. 7

分析：列表元素之间用逗号分隔，同时，一个列表也可以作为另一个列表的元素。需要注意的是，最后一个逗号后面并没有元素了(可省略)，所以列表 x 中共有 6 个元素，因此选 C 选项。

答案：C

7. Python 语句 sl=[4,5,6];s2=s1;s1[1]=0; print(s2)的运行结果是_____。

A. [4,5,6]　　B. [0,5,6]　　C. [4,0,6]　　D. 以上都不对

分析：在 Python 中的变量并不直接存储值，而是存储了值的内存地址或者引用，把一个列表变量赋值给另一个变量，这样两个变量指向同一个列表对象，对其中一个做任何修改都会立刻在另一个变量得到体现。该例中，s2=s1 就是 s2 和 s1 引用同一块内存，s1[1]=0 实际上 s2[1]也变为 0 了，所以选 C 选项。

答案：C

8. 已知列表 x=list(range(5))，那么执行语句 x.remove(3) 之后，表达式 x.index(4) 的值为_____。

A. 5　　B. 4　　C. 3　　D. 2

分析：x=list(range(5))，x=[0,1,2,3,4]，执行语句 x.remove(3)，则 x=[0,1,2,4]，x.index(4)则是查找元素 4 在列表中的下标，很明显下标值为 3(下标从 0 开始编号)，所以选 C 选项。

答案：C

9. 已知列表 x=[1, 3, 2]，执行语句 y=list(reversed(x)) 后，x 的值为_____。

A. [3,2,1]　　B. [1,3,2]　　C. [1,2,3]　　D. [2,3,1]

分析：内置函数 reversed()返回一个逆序后的对象(不改变原来列表 x 的值)，所以 y 的值为[2,3,1] ，而 x 并没有变化，因此选 B 选项。

答案：B

二、填空题

1. 列表对象的_____方法删除首次出现的指定元素，如果列表中不存在要删除的元素，则抛出异常。

分析：remove(x)在列表中删除第一个值为 x 的元素，该元素之后的所有元素前移并且索引减 1，如果列表中不存在 x 则抛出异常。

答案：remove( )

2. 已知列表对象 x=['11', '2', '3']，则表达式 max(x, key=len) 的值为_____。

分析：内置函数 max(x)用来求对象 x 的最大值，这里 x 是列表，求的是列表元素的最大值，同时 key=len 表示指定了比较大小的规则是元素的长度(注意，列表 x 的元素是字符串，字符串长度就是字符的个数)，长度最大的元素为'11'。

答案：'11'

3. 已知列表 x=list(range(10))，那么执行语句 del x[::2]之后，x 的值为_____。

分析：x=list(range(10))，则 x=[0,1,2,3,4,5,6,7,8,9]，x[::2]是切片操作，等价于 x[0:10:2]，即从列表第一个元素开始，以步长为 2(隔一个取一个)，获得下标为偶数的元

素，因此，del 删除的是列表 x 中偶数位置的元素，删除后 x=[1,3,5,7,9]。

答案：[1,3,5,7,9]

4. 假设列表对象 aList 的值为[3，4，5，6，7，9，11，13，15，17]，那么切片 aList[3:7]得到的值是_____。

分析：aList[3:7]，该切片操作没有第三个参数步长，取默认值 1，将 aList 中从下标 3 开始到下标 7(不包括)的元素依次获取组成新的列表[6,7,9,11]。

答案：[6,7,9,11]

5. 已知列表 x=[1.0，2.0，3.0]，那么表达式 sum(x)/len(x) 的值为_____。

分析：sum()和 len()都是内置函数，sum(x)求出列表 x 所有元素之和，为 6.0，len(x)求出列表 x 的元素个数，为 3，sum(x)/len(x)=6.0/3=2.0。

答案：2.0

6. 已知列表 x=[1，3，2]，那么执行语句 x.reverse() 之后，x 的值为_____。

分析：列表对象的 reverse()方法对列表所有元素进行原地(即用处理后的数据替换原来的数据)逆序，首尾交换，因此 x 变为[2,3,1]。

答案：[2，3，1]

7. 已知列表 x=[1，2]，那么执行语句 x.extend([3,4]) 之后，x 的值为_____。

分析：列表对象的 extend()方法将其参数(也是列表)的所有元素追加至原列表的尾部，因此，x 变为[1,2,3,4]。

答案：[1,2,3,4]

8. 已知列表 x=[1，2，3]，那么执行语句 x.insert(1，4) 之后，x 的值为_____。

分析：x.insert(1,4)表示在列表 x 的下标为 1 的位置插入新的元素 4，原下标位置 1 后面的元素依次后移，因此 x 变为[1,4,2,3]。

答案：[1,4,2,3]

9. 使用列表推导式生成包含 10 个数字 5 的列表，语句可以写为_____。

分析：只需要循环生成 10 次 5 即可用生成包含 10 个数字 5 的列表，可以写成[5 for i in range(10)]。

答案：[5 for i in range(10)]

10. aList=[－2，1，3，－6]，如何实现以绝对值大小从小到大将 aList 中内容排序_____。

分析：列表的排序可以调用其 sort()方法，sort()方法排序时可以指定排序的关键字为求绝对值 abs，此语句可写成 aList.sort(key=abs)。

答案：aList.sort(key=abs)

5.3　测　试　题

一、选择题

1. 已知 x=[1，2]和 y=[3，4]，那么 x+y 的结果是_____。

A. 3　　B. 7　　C. [1，2，3，4]　　D. [4，6]

2. 已知 x=[1, 2, 3],那么 x * 3 的值为_____。

A. 6　　B. [1, 6, 9]

C. [3, 6, 9]　　D. [1, 2, 3, 1, 2, 3, 1, 2, 3]

3. 已知 x=[1, 2, 3],执行语句 x.append(4)之后,x 的值是_____。

A. [1, 2, 3, 4]　　B. [4]　　C. [1, 2, 3]　　D. 4

4. 已知 x=[1, 2, 3, 4, 5, 6, 7],那么 x.pop()的结果是_____。

A. 1　　B. 4　　C. 7　　D. 5

5. sum([i * i for i in range(3)])的计算结果是_____。

A. 3　　B. 5　　C. 2　　D. 14

6. 下面代码输出结果为_____。

```
#1. x=[3]
#2. L=[3,2,1]
#3. for i in range(x[0]):
#4.       L.append(i*i)
#5. print(L)
```

A. [3, 2, 1, 0, 1, 4]　　B. [3,2,1,1,4,9]

C. [0,1,4]　　D. [1,4,9]

7. 已知列表 x=[1, 2],那么连续执行命令 y=x 和 y.append(3)之后,x 的值为_____。

A. [1,2]　　B. [1,2,3]　　C. [3,1,2]　　D. 不确定

8. 已知列表 x=[1, 3, 2],那么执行语句 a,b,c=sorted(x) 之后,b 的值为_____。

A. 1　　B. 2　　C. 3　　D. 不确定

9. 已知列表 x=[1, 2],那么执行语句 x.append([3]) 之后,x 的值为_____。

A. [1,2,3]　　B. [1,2]　　C. [3,1,2]　　D. [1,2,[3]]

10. 已知列表 x=[1, 3, 2],那么执行语句 x=x.reverse() 之后,x 的值为_____。

A. None　　B. [2,3,1]　　C. [1,3,2]　　D. [1,2,3]

11. 列表变量 ls 共包含 10 个元素,ls 索引的取值范围是_____。

A. −1～−9(含)的整数　　B. 0～10(含)的整数

C. 1～10(含)的整数　　D. 0～9(含)的整数

12. 以下不是 Python 复合数据类型的是_____。

A. 字符串类型　　B. 数组类型　　C. 列表类型　　D. 元组类型

13. 对于列表 s,能够返回列表 s 中第 i～j 且以 k 为步长的子列表的表达式是_____。

A. s[i:j:k]　　B. s[i,j,k]　　C. s[i;j;k]　　D. s(i,j,k)

14. 对于序列 s,以下选项对 min(s)描述正确的是_____。

A. 一定能够返回序列 s 中的最小元素

B. 可以返回序列 s 中的最小元素,但要求 s 中元素可比较

C. 可以返回序列 s 中的最小元素,如果存在多个相同的最小元素,则返回一个元组类型

D. 可以返回序列 s 中的最小元素,如果存在多个相同的最小元素,则返回一个列表类型

15. 以下程序输出的结果是_____。

```
#1. ls = [1,2,3]
#2. lt = [4,5,6]
#3. print(ls + lt)
```

A. [5,7,9]　　B. [1,2,3,[4,5,6]]
C. [1,2,3,4,5,6]　　D. [4,5,6]

16. 以下程序输出的结果是_____。

```
#1. a = [3,2,1]
#2. for i in a[::-1]:
#3.     print(i, end = ' ')
```

A. 1,2,3　　B. 3 2 1　　C. 1 2 3　　D. 3,2,1

17. 以下程序的输出结果是_____。

```
#1. a = [3,2,1]
#2. b = a[:]
#3. print(b)
```

A. [3,2,1]　　B. [ ]　　C. [1,2,3]　　D. 0xA1F8

18. 以下关于 Python 列表的描述中,错误的是_____。

A. 列表的长度和内容都可以改变,但元素类型必须相同
B. 可以对列表进行成员关系操作、长度计算和切片
C. 列表可以同时使用正向递增序号和反向递减序号进行索引
D. 可以使用比较操作符(如>或<等)对列表进行比较

19. 以下用来处理 Python 列表的方法中,错误的是_____。

A. interleave　　B. append　　C. insert　　D. remove

20. 以下代码的输出结果是_____。

```
#1. ls = [ 'book ', 23, [2010, 'stud1'], 20]
#2. print(ls[2][1][-1])
```

A. s　　B. stud1　　C. 1　　D. 结果错误

21. 以下代码的输出结果是_____。

```
#1. a = [[1,2,3], [4,5,6], [7,8,9]]
#2. s = 0
#3. for c in a:
#4.     for j in range(3):
#5.         s += c[j]
#6. print(s)
```

A. 6　　B. 0　　C. 24　　D. 45

22. 以下代码的输出结果是_____。

```
#1. vlist = list(range(5))
#2. print(vlist)
```

A. 0；1；2；3；4；　　B. 0 1 2 3 4

C. 0，1，2，3，4　　D. [0，1，2，3，4]

23. 以下关于列表变量 ls 操作描述中，错误的是_____。

A. ls.reverse()：反转列表 ls 中的所有元素

B. ls.clear()：删除 ls 的最后一个元素

C. ls.copy()：生成一个新列表，复制 ls 的所有元素

D. ls.append(x)：在 ls 最后增加一个元素

24. 列表 listV＝list(range(10))，以下能够输出列表 listV 中最大元素的是_____。

A. print(listV.reverse(i)[0])　　B. print(listV.max())

C. print(max(listV()))　　D. print(max(listV))

25. 以下代码的输出结果是_____。

```
#1. ls=[]
#2. for m in 'AB':
#3.     for n in 'CD ':
#4.         ls.append(m+n)
#5. print(ls)
```

A. ABCD　　B. AABBCCDD

C. ACADBCBD　　D. ['AC ', 'AD ', 'BC', 'BD']

26. 以下描述中，错误的是_____。

A. Python 语言通过索引来访问列表中元素，索引可以是负整数

B. 列表用方括号来定义，继承了序列类型的属性和方法

C. Python 列表是各种类型数据的集合，列表中的元素不能够被修改

D. Python 语言的列表类型能够包含其他的组合数据类型

27. 以下代码的输出结果是_____。

```
#1. s=[4, 2, 9, 1]
#2. s.insert(2, 3)
#3. print(s)
```

A. [4，2，9，2，1]　　B. [4，3，2，9，1]

C. [4，2，3，9，1]　　D. [4，2，9，1，3]

28. 以下代码的输出结果是_____。

```
#1. ls=[[1, 2,3], [[4, 5], 6], [7, 8]]
#2. print(len(ls))
```

A. 3　　B. 1　　C. 4　　D. 8

29. 以下代码的输出结果是_____。

```
#1. ls=["2020", "20.20", "Python"]
#2. ls.append(2020)
#3. ls.append([2020, "2020"])
#4. print(ls)
```

A. ['2020', '20.20', 'Python', 2020]

B. ['2020', '20.20', 'Python', 2020, [2020, '2020']]
C. ['2020', '20.20', 'Python', 2020, 2020, '2020']
D. ['2020', '20.20', 'Python', 2020, ['2020']]

30. 以下代码的输出结果是_____。

```
#1. l1 = ["aa", [2, 3, 3.0]]
#2. print(l1.index(2))
```

A. 2　　B. 3.0　　C. 3　　D. ValueError

31. 以下程序的输出结果是_____。

```
#1. ls1 = [1, 2, 3, 4, 5]
#2. ls2 = ls1
#3. ls2.reverse()
#4. print(ls1)
```

A. [5, 4, 3, 2, 1]　　B. [1, 2, 3, 4, 5]
C. 5, 4, 3, 2, 1　　D. 1, 2, 3, 4, 5

32. 以下程序的输出结果是_____。

```
#1. ls = [12, 44, 23, 46]
#2. for i in ls:
#3.     if i == '44':
#4.         print('found it! i = ',i)
#5.         break
#6. else:
#7.     print('not found it… ')
```

A. not found it…　　B. found it! i=44
C. found it! i=44
not found it…　　D. found it! i='44'
not found it…

33. 以下程序的输出结果是_____。

```
#1. L1 = [1, 2, 3, 4]
#2. L2 = L1.copy()
#3. L2.reverse()
#4. print(L1)
```

A. 1, 2, 3, 4　　B. [4, 3, 2, 1]
C. 4, 3, 2, 1　　D. [1, 2, 3, 4]

34. 以下程序的输出结果不可能的选项是_____。

```
#1. import random
#2. ls = [2, 3, 4, 6]
#3. s = 10
#4. k = random.randint(0,2)
#5. s += ls[k]
#6. print(s)
```

A. 12　　B. 14　　C. 13　　D. 16

35. 以下程序的输出结果是_____。

```
#1. ls=['绿茶', '乌龙茶', '红茶', '白茶', '黑茶']
#2. x='乌龙茶'
#3. print(ls.index(x,0))
```

A. 0　　B. 1　　C. －3　　D. －4

36. 二维列表 ls=[[9, 8], [7, 6], [5, 4], [3, 2], [1, 0]],能够获得数字 4 的选项是_____。

A. ls[－2,0]　　B. ls[3,2]　　C. ls[2,2]　　D. ls[－3,－1]

37. 以下程序的输出结果是_____。

```
#1. lt=['绿茶', '乌龙茶', '红茶', '白茶', '黑茶']
#2. ls=lt
#3. ls.clear()
#4. print(lt)
```

A. 变量未定义的错误

B. []

C. ['绿茶', '乌龙茶', '红茶', '白茶', '黑茶']

D. '绿茶', '乌龙茶', '红茶', '白茶', '黑茶'

二、填空题

1. 列表对象的 sort()方法用来对列表元素进行原地排序,该方法的返回值为_____。

2. 表达式"[3] in [1,2,3,4]"的值为_____。

3. 表达式 list(range(1, 10, 3))的值为_____。

4. 已知列表对象 x=['15', '25', '3'],则表达式 max(x) 的值为_____。

5. 已知列表 x 中包含超过 5 个以上的元素,那么表达式 x == x[:5]+x[5:] 的值为_____。

6. 已知列表 x=[1, 2, 3],那么表达式 [value for index, value in enumerate(x) if index==2] 的值为_____。

7. 已知列表 x=[1, 3, 2],那么执行语句 y=list(reversed(x)) 之后,y 的值为_____。

8. 已知列表 x=[1, 3, 2],那么执行语句 a, b, c=map(str, sorted(x)) 之后,c 的值为_____。

9. 已知列表 x=[1, 2],那么连续执行命令 y=x[:] 和 y.append(3) 之后,x 的值为_____。

10. _____(可以、不可以)使用 del 命令来删除元组中的部分元素。

11. 已知列表 x=[1, 2, 3],那么执行语句 x.pop(0) 之后,x 的值为_____。

12. 假设有一个列表 a,现要求从列表 a 中每 3 个元素取 1 个,并且将取到的元素组成新的列表 b,可以使用语句_____。

13. 已知 x=(2,),那么表达式 x * 3 的值为_____。

14. 已知列表 x=[1, 2, 3],那么表达式 list(enumerate(x)) 的值为_____。

15. 写一行代码实现对列表 a 中的偶数位置的元素进行加 3 后求和，可以写成_____。

16. 表达式 [i for i in range(10) if i>8] 的值为_____。

17. 表达式 [5 for i in range(3)] 的值为_____。

18. 表达式 [1, 2, 3].count(4) 的值为_____。

19. 假设列表对象 aList 的值为[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]，那么切片 aList[3:7] 得到的值是_____。

20. 已知 x=[3, 7, 5]，那么执行语句 x.sort(reverse=True) 之后，x 的值为_____。

21. 已知 x=[3, 5, 7]，那么表达式 x[10:]的值为_____。

22. 已知 x=[3, 5, 7]，那么执行语句 x[:3]=[2]之后，x 的值为_____。

23. 已知 x=[1, 2, 3, 4, 5]，那么执行语句 del x[1:3] 之后，x 的值为_____。

24. 已知 x=[1, 2, 1]，那么表达式 id(x[0]) == id(x[2]) 的值为_____。

25. 表达式 [index for index, value in enumerate([3, 5, 7, 3, 7]) if value == max([3,5,7,3,7])] 的值为_____。

三、编程题

1. 编写程序，求列表中的元素个数，最大值、最小值、元素之和、平均值，并请思考，有哪几种实现方法？至少用两种方法实现。

2. 用户输入若干个分数存放到列表中，求所有分数的平均分。每输入一个分数后询问是否继续输入下一个分数，回答“yes”就继续输入下一个分数，回答“no”就停止输入分数。

3. 将一个列表中的元素为奇数的放到前面，偶数放到后面，并输出。

4. 编写程序，生成 20 个 0～100 随机数的列表，然后将列表中的奇数放到列表的前面，偶数放到后面，且奇数和偶数都要按升序存放，统计奇数和偶数的个数，并输出结果。

5. 统计 1900—2019 年中有多少闰年(使用列表推导式)。

6. 有一个 4 行 4 列的矩阵，求出其转置矩阵。

7. 将一个 4 行 4 列的矩阵逆时针旋转 90°。

8. 猴子吃桃问题：猴子第一天摘下若干个桃子，当天吃了一半，还不过瘾，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加一个。到第十天早上再想吃时，发现就只剩一个桃子了。求第一天共摘了多少个桃子？

9. 编写程序，接收一个数字的列表；计算得到一个新的列表，其中，第 i 个元素是原列表的前 i 个元素的和。例如，原列表 x=[1,2,3,4,5]，则新列表为[1,3,6,10,15]。

10. 在歌星大奖赛中，有 10 个评委为参赛的选手打分，分数为 1～100 分。选手最后得分为：去掉一个最高分和一个最低分后其余 8 个分数的平均值。请编写一个程序实现。

5.4　实 验 案 例

一、输出列表的子列表

1. 实验要求

用户输入一个列表和两个整数作为下标，然后输出列表中介于两个下标之间的元素组

成的子列表。例如,用户输入[1,2,3,4,5,6]和 2,5,程序输出[3,4,5,6]。编写并输入代码,保存到程序文件 Ex5-1.py 中,运行程序并观察结果。

2. 算法分析

使用列表的切片操作即可求出子列表。

3. 完善程序

```
#1. x = input('Please input a list:')
#2. x = eval(x)
#3. start, end = eval(input('Please input the start position and the end position:'))
#4. print(_____)
```

4. 调试程序

保存文件为 Ex5-1.py,运行程序,程序输出结果如下。

```
Please input a list:[1,2,3,4,5,6]
Please input the start position and the end position:2,5
[3, 4, 5, 6]
```

二、序列元素排序

1. 实验要求

生成包含 20 个随机数的列表,然后将前 10 个元素升序排列,后 10 个元素降序排列,并输出结果。编写并输入代码,保存到程序文件 Ex5-2.py 中,运行程序并观察结果。

2. 算法分析

使用内置函数 sorted()可对列表排序。

3. 完善程序

```
#1. import random
#2. #随机生成 20 个范围在[0,100]的随机数
#3. x = [random.randint(0,100) for i in range(20)]
#4. print(x)       #输出排序前的列表
#5. x[:10] = _____
#6. _____ = sorted(x[10:], reverse = True)
#7. print(x)       #输出排序后的列表
```

三、删除指定的元素

1. 实验要求

生成一个包含 50 个随机整数的列表,然后删除其中所有偶数。编写并输入代码,保存到程序文件 Ex5-3.py 中,运行程序并观察结果。

2. 算法分析

循环遍历列表元素,判断为偶数则将其删除。

3. 完善程序

```
#1. import random
#2. x = [random.randint(0,100) for i in range(50)]
#3. print(x)
#4. for i in range(len(x))[::-1]:          #从后往前遍历
```

```
#5.      if ________:
#6.          del x[i]
#7. print(x)
```

四、部分元素排序

1. 实验要求

生成一个包含 20 个随机整数的列表，然后对其中偶数下标的元素进行降序排列，奇数下标的元素不变。编写并输入代码，保存到程序文件 Ex5-4.py 中，运行程序并观察结果。

2. 算法分析

使用切片完成，切片的步长为 2。

3. 完善程序

```
#1. import random
#2. x = [random.randint(0,100) for i in range(20)]
#3. print(x)
#4. x[::2] = sorted(________)
#5. print(x)
```

五、整数的因式分解

1. 实验要求

用户从键盘输入小于 1000 的整数，对其进行因式分解。例如，10＝2×5，60＝2×2×3×5。编写并输入代码，保存到程序文件 Ex5-5.py 中，运行程序并观察结果。

2. 算法分析

对于一个整数 x，从 i(i 初始值为 2)开始判断，如果 x 能被 2 整除，则认为 2 是一个有效因子，将其保存在列表中，然后 x 的值变为 x//2，如果 x 不能被 2 整除，则开始从下一个数 i＋1 判断。如此反复，直到最后 x 值变为 1，保存在列表中的所有数相乘就是因式分解的结果，其流程图如图 5-1 所示。

3. 完善程序

```
#1. x = input('Please input an integer less than 1000:')
#2. x = eval(x)
#3. t = x
#4. i = 2
#5. result = []                    #该列表用于存放 x 的所有因子
#6. while _____________:
#7.     if t % i == 0:
#8.         _____________
#9.         t = _____________
#10.     else:
#11.         _____________
#12. print(x,' = ',end = '')
#13. for i in range(len(result)):      #输出因式分解的结果
#14.     if _____________
#15.         print(result[i],end = '')
#16.     else:
#17.         print(result[i],' * ',end = '')
```

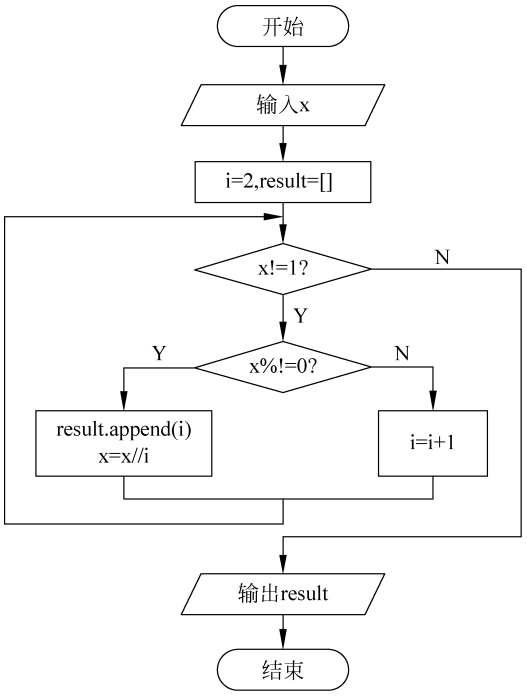

图 5-1　因式分解

六、删除重复元素

1. 实验要求

将列表中的重复元素去除，例如，列表内容为[1,2,3,1,5,3,7]，去除重复元素后列表内容为[1,2,3,5,7]。编写并输入代码，分别保存到程序文件 Ex5-6-1.py 和 Ex5-6-2.py 中，运行程序并观察结果。

2. 完善程序

方法一：

```
#1. l=[1,2,3,1,5,3,7]          #将本方法代码存入程序文件 Ex5-6-1.py
#2. L=[]
#3. for i in l:
#4.     if i not in L:
#5.         _____________
#6. print(L)
```

方法二：使用集合(第 6 章会学到集合)

```
#1. l=[1,2,3,1,5,3,7]          #将本方法代码存入程序文件 Ex5-6-2.py
#2. list(set(l))               #直接将列表转换为集合,再转换为列表
```

七、插入排序法

1. 实验要求

不使用内置函数和列表 sort()方法的前提下，使用插入排序法对列表中元素排序。编写并输入代码，保存到程序文件 Ex5-7.py 中，运行程序并观察结果。

2. 算法分析

插入排序是最常见的数据排序方法之一，它的基本原理就是将一个数据插入到一组已经排好序的序列当中，插入后该序列依旧是有序的。

算法思想(从小到大排序)：

(1) 一组数据(包含 N 个元素的列表 x)，设置变量 i,j,temp,i 是待插入元素的下标，将 x[i]赋值给 temp,j 用来搜索比较。

(2) i 的初始值为 1，即从第二个元素开始，只有一个元素的列表直接被认为是已经排好序的，所以从第二个元素开始，j 的初始值为 i－1。

(3) 将 i 之前的数据都与 temp 比较，即与待插入数据进行比较，用 j 来移动下标，每次 j=j－1，直到找到一个比 temp 小的或等于的值，亦或找到了该序列的尽头(即下标为 0)停止，将 temp 放到 j＋1 的位置，即将待插入的数据插入到 j＋1 的位置，该待插入数据就已经插入到了正确的位置。

(4) 然后将 i=2，从第三个元素开始，j=i－1，执行 temp=a[i]，重复步骤 3，直到 i==N－1 停止，即所有数据都已经放到了正确的位置上，该序列即为一个有序序列。

例如，“17、13、9、12、5”的排序过程如表 5-2 所示。

表 5-2　插入排序过程示例

原始数据	17	13	9	12	5
第一轮	**13**	17	9	12	5
第二轮	**9**	13	17	12	5
第三轮	9	**12**	13	17	5
第四轮	**5**	9	12	13	17

3. 完善程序

```
#1. x = input('Please input a list:')
#2. x = eval(x)
#3. n = len(x)
#4. print("排序前: ",x)
#5. for i in range(1,n):              #外循环(1～n－1)
#6.     if _____________:
#7.         temp = x[i]
#8.         j = i－1
#9.         while _____________:    #内循环,元素后移
#10.            x[j＋1] = x[j]
#11.            j－= 1
#12.        _____________
#13. print("\n 排序后: ",x)
```

八、选择排序

1. 实验要求

不使用内置函数和列表 sort()方法的前提下，使用选择排序法对列表中元素排序。编写并输入代码，保存到程序文件 Ex5-8.py 中，运行程序并观察结果。

2. 算法分析

选择法排序的思路是：将 n 个数依次比较，保存最大数的下标位置，然后将最大数和第

1 个列表元素换位；接着再将 n－1 个数依次比较，保存次大数的下标位置，然后将次大数和第 2 个列表元素换位；接着再将 n－2 个数依次比较，保存第 3 大数的下标位置，然后将第 3 大数和第 3 个列表元素换位。按此规律，直至比较换位完毕。

例如，“2、3、9、6、5”的排序过程示意如下。

第 1 步：[2　3　9　6　5]5 个数比较，保存最大数 9 的下标位置 2，将 9 和第 1 个数组元素 2 换位。

第 2 步：**9**　[3　2　6　5]余下 4 个数比较，保存次大数 6 的下标位置 3，将 6 和第 2 个数组元素 3 换位。

第 3 步：**9　6**　[2　3　5]余下 3 个数比较，保存第 3 大数 5 的下标位置 4，将 5 和第 3 个数组元素 2 换位。

第 4 步：**9　6　5**　[3　2]最后两个数比较，不换位。

至此，排序完成。

3. 完善程序

```
#1. x = input('Please input a list:')
#2. x = eval(x)
#3. n = len(x)
#4. for j in range(n):                    #外循环(0～n－1)
#5.       max1 = x[j]
#6.       label = j                       #当前位置下标
#7.       for i in range(j + 1,n):        #内循环,查找最大值下标
#8.           if _____________:
#9.                max1 = x[i]
#10.               _____________
#11.      t = x[label]                    #元素交换
#12.      _____________
#13.      x[j] = t
#14. print('排序结果：', x)
```

九、序列处理

1. 实验要求

输入三个正数序列，从三个序列中各取一个值相乘，输出乘积最大的三个数及各序列中所取数的位置。例如，输入的三个序列为[2,3,1]，[4,6,3]，[5,2,4]，最大值为 3×6×5，下标 1,1,0 组合，第一个序列第二个，第二个序列第二个，第三个序列第一个。编写并输入代码，保存到程序文件 Ex5-9.py 中，运行程序并观察结果。

2. 算法分析

分别取出三个序列的最大元素，相乘的结果即为最大值，同时记录下所取元素在各自序列中的下标即可。

3. 完善程序

```
#1. x = input('Please input a list:')
#2. x = eval(x)
#3. y = input('Please input a list:')
#4. y = eval(y)
#5. z = input('Please input a list:')
```

```
#6. z = eval(z)
#7. i = _____
#8. iPos = _____
#9. j = max(y)                    #列表 y 的最大值
#10.jPos = y.index(j)             #最大值所在的下标
#11.k = max(z)
#12.kPos = z.index(k)
#13.result = _____
#14.print(result, (iPos, jPos, kPos))
```

4. 思考题

如果序列中包括负数，程序需要调整吗？为什么？如果需要调整，应该怎么改动？

十、水仙花数

1. 实验要求

水仙花数是指 1 个 3 位的十进制数，其各位数字的立方和等于该数本身。例如，153 是水仙花数，因为 $153=1^3+5^3+3^3$。编写并输入代码，保存到相应程序文件中，运行程序并观察结果。

2. 算法分析

方法很多，例如：

(1) 第 4 章介绍过用“穷举法”分别搜索 100～999 满足条件的值，采用算术运算符求出个位、十位、百位。

(2) 可以使用 Python 内置函数 divmod()依次求出个位、十位、百位。

(3) 可以使用序列解包。

(4) 可以使用 lambda 表达式。

3. 完善程序

参考代码一，算术运算符。代码保存到程序文件 Ex5-10-1.py 中。

```
#1. for num in range(100, 1000):
#2.     bai = num // 100          #百位
#3.     shi = _____               #十位
#4.     ge = num % 10             #个位
#5.     if _____:                 #判断是否水仙花数
#6.         print(num)
```

参考代码二，内置函数。代码保存到程序文件 Ex5-10-2.py 中。

```
#1. for num in range(100, 1000):
#2.     bai, rest = divmod(num, 100)     #百位及余数
#3.     shi, ge = _____
#4.     if ge**3 + shi**3 + bai**3 == num:
#5.         print(num)
```

参考代码三，序列解包。代码保存到程序文件 Ex5-10-3.py 中。

```
#1. for num in range(100, 1000):
#2.     bai, shi, ge = map(_____)
#3.     if ge**3 + shi**3 + bai**3 == num:
#4.         print(num)
```

参考代码四，lambda 表达式。代码保存到程序文件 Ex5-10-4.py 中。

```
#1. for num in range(100, 1000):
#2.     r = map(_____________)
#3.     if sum(r) == num:
#4.         print(num)
```

十一、查找特定元素

1. 实验要求

随机生成一个列表，查找列表元素中最大值的所有出现和首次出现位置。例如，[23,45,28,56,45,56,32,11,21,10]，其最大值为 56，出现的位置有 3 和 5，首次出现的位置是 3。编写并输入代码，保存到程序文件 Ex5-11.py 中，运行程序并观察结果。

2. 算法分析

可以利用内置函数 max 求出列表元素的最大值，再遍历列表元素，将所有值等于该最大值的元素下标记录到另一个列表中。

3. 完善程序

```
#1. import random
#2. randomNums = [random.randrange(30) for i in range(10)]
#3. m = max(randomNums)          #最大值
#4. listPos = []                 #listPos 用于存放 m 在列表中出现的所有位置下标
#5. length = _____________
#6. for i in range(length):
#7.     if(randomNums[i] == m):
#8.             _____________
#9. print("序列：", randomNums)
#10.print("最大值为：", m)
#11.print("序列中出现该最大值的位置有：", listPos)
#12.print("其中，首次出现的位置为：", listPos[0])
```

十二、约瑟夫问题

1. 实验要求

给 10 个学生编号 1～10，按顺序围成一圈，按 1～3 报数，凡报到 3 者出列，然后下一个人继续从 1 开始报数，直到最后只剩下一个人，计算剩下这个人是第几号学生并输出。编写并输入代码，保存到程序文件 Ex5-12.py 中，运行程序并观察结果。

2. 算法分析

定义一个学生编号的列表 listNo，里面存放 10 个值都为 1 的元素，列表元素的下标即表示学生的编号，例如，下标 0 表示的是 1 号同学，下标 1 表示的是 2 号同学，……，下标 9 表示的是 10 号同学。报数的过程就是将列表中对应的元素相加的过程，当加到和为 3 时，说明该同学应该出列，则将该元素值变为 0，同时将学生总数减去 1，直到最后学生总数为 1 为止。

3. 完善程序

```
#1. num = int(input("请输入学生总数："))
#2. listNo = []
#3. j = 0
```

```
#4. for i in range(num):
#5.       listNo.append(1)          #将列表元素全部初始化为1
#6. sum1 = 0
#7. while True:
#8.       sum1 += listNo[j]
#9.       if _____________:
#10.            listNo[j] = 0       #表示下标为j的同学出列
#11.            sum1 = 0            #重新报数,累加结果
#12.            #print(listNo)      #跟踪测试,观察中间有同学出列后列表变化情况
#13.            num -= 1            #有同学出列,num - 1
#14.            if _____________:   #循环强行结束
#15.                  break
#16.      _____________
#17.      if j >= len(listNo):      #到达列表最后的元素后,重新回到列表开头
#18.            _____________
#19.for i in range(len(listNo)):
#20.      if _____________:
#21.            print("最后剩下的同学是:",i + 1,"号!")
#22.            break;
```