第 1 章

1.1 Python 安装扩展库常用的工具是__pip_____和 conda，其中后者需要安装 Python 集成开发环境 Anaconda3 之后才可以使用。

1.2 Python 程序文件扩展名主要有___py_____和____pyw____两种，其中后者常用于 GUI 程序。

1.3 多选题：下面属于 Python 语言特点的有哪些？ABCD

A. 开源　　B. 免费　　C. 跨平台　　D. 解释执行

1.4 多选题：下面导入标准库对象的语句中正确的有？ABC

A. from math import sin　　B. from random import random

C. from math import *　　D. import *

1.5 多选题：下面描述中符合 Python 语言的有哪些？ABCD

A. 跨平台　　B. 开源　　C. 解释执行　　D. 支持函数式编程

1.6 多选题：下面描述中符合 Python 语言的有哪些？ABCD

A. 跨平台　　B. 开源　　C. 免费　　D. 扩展库丰富

1.7 多选题：下面能够支持 Python 程序开发和运行的环境有哪些？ABCD

A. IDLE　　B. Anaconda3　　C. PyCharm　　D. Eclipse

1.8 多选题：下面能够支持 Python 程序开发和运行的环境有哪些？CD

A. Word　　B. 记事本　　C. VS Code　　D. wingIDE

1.9 多选题：下面哪些是正确的 Python 标准库对象导入方式？BD

A. import math.sin　　B. from math import sin

C. import math.*　　D. from math import *

1.10 判断对错：安装 Python 扩展库时只能使用 pip 工具联网在线安装，如果安装不成功就没有别的办法了。（错）

1.11 判断对错：不管计算机上安装了多少个 Python 的版本，每个扩展库只需要安装一次，就可以在所有 Python 版本的开发环境中使用了。（错）

1.12 判断对错：使用 pip 工具安装扩展库时，可以切换到 cmd 或 Powershell 环境中运行命令，也可以在 IDLE 交互模式下直接运行命令。（错）

1.13 判断对错：使用 pip 工具安装扩展库时，如果文件下载速度非常慢，可以使用-i 选项指定从国内服务器上下载和安装，大幅度提高速度。

1.14 判断对错：使用命令 pip freeze 可以查看本机当前 Python 版本下已安装的所有扩展库名称和版本号。

1.15 判断对错：使用命令 pip freeze > requirements.txt 可以把本机当前 Python 版本下已安装的扩展库名称和版本信息写入文件 requirements.txt 中。

1.16 判断对错：如果只需要 math 模块中的 sin()函数，建议使用 from math import sin 来导入，不推荐使用 import math 导入整个模块然后再使用 math.sin()的方式进行调用。

1.17 判断对错：内置对象在程序中任何位置都可以直接使用，不需要导入任何内置模块、标准库或扩展库。（错）

1.18 判断对错：Python 官方安装包没有包含任何扩展库，只有内置对象、内置模块和标准库，这些是 Python 自带的，不需要导入就可以直接使用。（错）

1.19 判断对错：在编写 Python 程序时，对代码进行缩进只是为了好看，不缩进也不影响程序的正常执行。（错）

1.20 判断对错：在编写程序时，不管一条语句有多长，都必须写在同一行中，不能换

行。（错）

1.23 Python 程序里面__name__属性作用？

每个 Python 程序在运行时都有一个__name__属性，利用__name__属性的值可以识别和控制 Python 程序的运行方式：1）如果程序作为模块被导入，则其__name__属性的值被自动设置为模块文件名；2）如果作为程序直接运行，则其__name__属性值被自动设置为字符串'__main__'。

上机要求 1：主要是 IDLE 和 Anaconda3 的安装，以及 packages 的安装（P13：表 1-1）。

第 2 章

2.1 表达式 int('11111', 2)的值为___31_____。

2.2 ord();chr()的使用

2.3 表达式 3**2+5 的值为___14_____。

2.4 表达式 3==3 is not True 的值为____True_____。

(3==3) and (3 is not True) 3<5<8 (3<5) and (5<8)

测试运算符（in；not in;is;is not）和关系运算符具有相同的优先级

括号和属性优先级别最高；算术运算 > 位运算 > 比较运算 > 逻辑运算；赋值最后

2.5 表达式 (3==3) is not True 的值为___False_____。

2.6 表达式 3 in [1, 2, 3, 4] 的值为_____True_____。

2.7 表达式 3 not in [1, 2, 3, 4] 的值为____False_____。

2.8 Python 关键字中用来判断两个对象的引用是否相同的是_____is_____。

2.9 已知变量 value 的值为 3，那么表达式 value%2 的值为___1_____。

2.10 已知变量 value 的值为 7，那么表达式 value%5 的值为____2_____。

2.11 已知集合 A 是集合 B 的真子集，那么表达式 A < B 的值为____True_____。

2.12 Python 关键字_____del_____既可以删除列表（字典）中的一个元素，也可以删除整个列表。

2.13 已知 x = 3，那么执行语句 x += 6 之后，x 的值为__9_____。

注意对于列表的进行复合加+=运算和普通加+运算的区别 （3.1.4 节）

2.14 表达式 13 // 4 的值为___3_____。

2.15 表达式 -13 // 4 的值为____-4_____。

注意是向下取整

2.16 表达式 -10 % 4 的值为____2_____。

-10-（-10//4*4）= 2

2.17 表达式 10 % -3 的值为_____-2_____。

-10-（10//（-3）*（-3））= -2

注意结果符号和除数相同

2.18 表达式 3>5 and a<b 的值为____False_____。

2.19 表达式 bool(range(8,5)) 的值为_____False_____。

2.20 表达式 sum(map(int, str(123456))) 的值为_____21____。

2.21 表达式 eval('3*2'+'22') 的值为_____666____。

2.22 表达式 max(['121', '34']) 的值为_____'34'_____。

2.23 单选题：表达式[1, 2, 3] * 2 的值为？ A

A. [1, 2, 3, 1, 2, 3]　　B. [2, 4, 6]

C. [1, 1, 2, 2, 3, 3]　　D. 无法计算

2.24 单选题：表达式 68 // -7 的值为？ A

A. -10　　B. 10　　C. 9　　D. -9

2.25 单选题：表达式 {1,2,3,4} - {2,4,5} 的值为？A

A. {1,3}　　B. {5}　　C. {1,3,5}　　D. 错误表达式，无法计算

2.26 单选题：表达式 3 and 5 的值为？B

A. 3　　B. 5　　C. 8　　D. True

2.27 单选题：表达式 3 or 5 的值为？A

A. 3　　B. 5　　C. 8　　D. True

2.28 单选题：表达式 3 in range(1, 20, 4) 的值为？B

A. True　　B. False　　C. 1　　D. 0

2.29 单选题：表达式 bin(int('11',16)) 的值为？C

A. 11　　B. 17　　C. '0b10001'　　D. 10001

2.30 多选题：下面属于可哈希对象的有？AB

A. 345　　B. '0b10001'　　C. [1, 2, 3]　　D. {3}

2.31 多选题：下面可以作为内置函数 reversed()的参数的有？AB

A. 列表　　B. 元组　　C. map 对象　　D. zip 对象

2.32 单选题：下面不是合法变量名的有？C

A. age　　B. name　　C. 3_name　　D. height

变量必须以字母、汉字或者下划线开头，不能用关键字

2.33 单选题：下面不是合法变量名的有？C

A. width　　B. area　　C. for　　D. door_number

2.34 单选题：已知 x = [1, 2]和 y = [3, 4]，那么 x+y 的结果是？D

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

2.35 单选题：sum([i*i for i in range(3)])的计算结果是？B

A. 3　　B. 5　　C. 2　　D.14

2.36 单选题：下面运算符中可以用于不同内置类型对象（整数、实数、复数不做区分）之间运算的有哪些？ C

A. +　　B. -　　C. *　　D. /

2.37 多选题：下面属于 Python 内置对象的有哪些？

A. str　　B. list　　C. dict　　D. set

2.38 单选题：执行语句 d, _ = divmod(36, 12)之后，变量 d 的值为？

A. 36　　B. 12　　C. 3　　D. 语法错误无法执行

2.39 多选题：下面属于合法变量名的有哪些？

A. def　　B. while　　C. age　　D. name

2.40 多选题：下面属于合法变量名的有那些？

A. 3name　　B. name_3　　C. 姓名　　D. _name

2.41 多选题：下面属于合法数字的有哪些？

A. 0b1101　　B. 0o784　　C. 0xb2　　D. 789

2.42 多选题：下面属于合法数字的有那些？

A. 1_234_567　　B. 1e8　　C. 9.8　　D. .3

2.43 单选题：已知 x = [1, 2, 3]，那么 x*3 的值为？

A. 6　　B. 18

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

2.44 单选题：下面内置函数中，用来计算列表长度的函数是？
A. max()　　B. sum()　　C. int()　　D. len()
2.45 多选题：下面表达式的值为 True 的有那几个？
A. 5>3　　B. 3 and 5
C. 5==3　　D. 3 not in [1,2,5]
2.46 多选题：下面表达式作为条件表达式时表示条件成立的有那几个？
A. 5>3　　B. 3 and 5　　C. 5==3　　D. 3 not in [1,2,5]
2.47 单选题：表达式 max([1,2,3], [2,1,3,0], [3,2,1]) 的值为？
A. 3　　B. [3]　　C. [3, 2, 1]　　D. [2, 1, 3, 0]
2.48 单选题：表达式 max([1,2,3], [2,1,3,0], [3,2,1], key=len) 的值为？
A. [3, 2, 1]　　B. [2, 1, 3, 0]　　C. 3　　D. [3]
2.49 单选题：已知 x = ['aaaa', 'bc', 'd', 'b', 'ba']，那么表达式 sorted(x, key=len) 的值为？
A. ['d', 'b', 'bc', 'ba', 'aaaa']　　B. ['d', 'b', 'ba', 'bc', 'aaaa']
C. ['b', 'bc', 'ba', 'd', 'aaaa']　　D. ['b', 'ba', 'bc', 'd', 'aaaa']
2.50 表达式 len(zip([1,2,3], 'abcdefg'))的值为 3。错
len()不适用于惰性求值的生成器对象和 map，zip 等迭代器对象
2.51 语句 x = input(3) 的作用是自动输入 3 并赋值给变量 x。错
input(3)：显示提示符 3，等待用户输入，不会自动赋值 3。
只有用户主动输入后，x 才会被赋值（且类型为字符串）
2.52 在 Python 中可以使用 if 作为变量名。错
2.53 9999**9999 这样的表达式在 Python 中无法运行，因为计算结果太大了。错
2.54 已知 x='[1,2,3,4]'，那么表达式 list(x) 的值为 [1, 2, 3, 4]。错
2.55 对于两个集合 A 和 B，表达式 A-B 和 B-A 的值是一样的。错
2.56 加法运算符可以连接两个字典得到新字典。错
2.57 加法运算符可以连接两个列表得到新列表。对
2.58 3+4j 不是合法的 Python 表达式。错
2.59 3 不是合法的 Python 表达式。错
2.60 0o12f 是合法的八进制数字。错
2.61 0o1234567 是合法的八进制数字。对
2.62 在 Python 中 0xad 是合法的十六进制数字表示形式。对
2.63 4j 是合法 Python 数字类型。对
2.64 在 Python 中 0oa1 是合法的八进制数字表示形式。错
2.65 在 Python 程序中可以使用 for 作为变量名。错
2.66 在 Python 程序中可以使用 while 作为变量名。错
2.67 在 Python 程序中可以使用 id 作为变量名，但是不建议这样做。对
2.68 在 Python 程序中可以使用 max_ 作为变量名，并且不会有什么副作用。对

第 2 章要求熟悉和掌握内置对象及其特点、运算符、关键字和常用内置函数的运用

（min;max;sum;sorted;reversed;input;print;enumerate,map;reduce;filter;range;zip;eval）

from functools import reduce
a = ['1','2','3']

reduce(lambda x,y:x+y,a)

（1） 填空完成 mian()函数，要求接收一个正整数，输出该整数百位数，十位数和个位数上的数字的和，比如输入 123，输出为 1+2+3=6。可以多行实现，尽量只写一行（利用 map 函数）

```
def main(num):
    return sum(map(int,str(num)))
```

（2） 填空完成 mian()函数，要求接收一个包含若干整数的列表 lst，要求返回一个新列表，新列表 lst2 包含 lst 中的唯一元素（重复元素只保留一个），并且里面的元素按从大到小的顺序排列，例如调用函数 main([1,3,2,1,4])会输出[4,3,2,1]。提示：利用 set()元素不允许重合原理

```
def main(lst):
    lst_unique=list(set(lst))
    lst2 = sorted(lst_unique,reverse=True)
    return lst2
```

（3）接受一个列表，要求去掉列表里面重复的元素，并按照在列表中首次出现的顺序排序

```
a = [1,8,3,5,1,4,6,8]
sorted(set(a),key=a.index)
```

得到： [1, 8, 3, 5, 4, 6]

第 3 章

3.1、False 3.2、None （sort 和 sorted 的区别，reverse 和 reversed 区别）

3.3、remove() 3.4、[6, 7, 9, 11] 3.5、大括号、键、值、键

3.6、items()、keys()、values() 3.7、dict(zip(a,b)) 3.8、b = a[::3]

3.9、[5 for i in range(10] 3.10、不可以

3.11 已知 vec = [[1,2], [3,4]]，则表达式 [col for row in vec for col in row] 的值为___[1, 2, 3, 4]_____。

3.12 已知 vec = [[1,2], [3,4]]，则表达式 [[row[i] for row in vec] for i in range(len(vec[0]))] 的值为_____[[1, 3], [2, 4]]_____。

3.13 表达式 {1, 2, 3} < {3, 4, 5} 的值为___False_____。

3.14 表达式 [1, 2, 3].count(4) 的值为_____0_____。

3.15 表达式 [1, 2, 3, 1, 2].index(2) 的值为____1_____。

3.16 单选题：表达式 set().union([1,2,3], (4,5), {6,7}) 的值为？

A. {1, 2, 3, 4, 5, 6, 7} B. {1, 2, 3, 4, 5}

C. {4, 5, 6, 7} D. {4, 5}

3.17 单选题：已知 x = {1: 'a', 2: 'b', 3: 'c'} 和 y = {1, 3, 4}，那么表达式 x.keys() - y 的值为？

A. {2} B. {3} C. {1, 3} D. 表达式错误，无法计算

3.18 单选题：已知 x = {1: 3, 2: 1, 3: 1} 和 y = {1, 3, 4}，那么表达式 x.values() - y 的值为？

A. {2}　　B. {3}　　C. {1, 3}　　D. 表达式错误，无法计算

dict_keys 对象支持集合操作（如 -、&、|），因为它在功能上类似于集合（set）。

dict_values 对象是不可哈希的，不支持；但可以通过以下实现：

a = {6,7}

a.difference({1:2,3:4}.values())

a.difference((6, 8))

3.19 多选题：下面的列表方法中有返回值并且返回值不是空值 None 的有哪些？

A. sort()　　B. pop()　　C. index()　　D. reverse()

3.20 多选题：下面的列表方法中返回值一定为空值的有哪些？

A. sort()　　B. reverse()　　C. count()　　D. append()

3.21 多选题：下面的列表方法中返回值一定为空值的有哪些？

A. insert()　　B. clear()　　C. count()　　D. extend()

3.22 多选题：下面的列表方法中不影响列表内存地址的有那些？

A. sort()　　B. pop()　　C. index()　　D. reverse()

3.23 多选题：下面的列表方法中不影响列表中元素的有那些？

A. sort()　　B. count()　　C. index()　　D. pop()

3.24 多选题：下面的列表方法中返回值一定为空值 None 的有那些？

A. sort()　　B. index()　　C. pop()　　D. reverse()

3.25 多选题：下面的列表方法中返回值一定为空值 None 的有那些？

A. insert()　　B. copy()　　C. remove()　　D. append()

3.26 多选题：下面可以使用表示位置或序号的整数做下标访问其中元素的有哪些？

A. 列表　　B. 元组　　C. 字典　　D. 集合　　E. 字符串

3.27 多选题：下面可以支持下标运算的有那些？

A. 列表　　B. 元组　　C. 字典　　D. 集合　　E. 字符串

3.28 编写程序，生成包含 1000 个 0~100 的随机整数，并统计每个元素的出现次数。

3.29 编写程序，用户输入一个列表和两个整数作为下标，然后使用切片获取并输出列表中介于两个下标（都包含）之间的元素组成的子列表。

3.30 设计一个字典，并编写程序，用户输入内容作为“键”，然后输出字典中对应的“值”，如果用户输入的“键”不存在，则输出“您输入的键不存在！”。

3.31 编写程序，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，并输出排序之后的结果。

3.28

```
d = dict()
for v in x:
        d[v] = d.get(v,0) + 1
for k, v in d.items():
print(k, v, sep=':')
```

3.29

```
x = input('请输入一个列表：')
x = eval(x)
start = int(input('请输入开始位置：'))
```

```
end = int(input('请输入结束位置：'))
print(x[start:end+1])
```

3.30

```
d = {1:'a', 2:'b', 3:'c', 4:'d'}
v = input('请输入一个键：')
v = eval(v)
print(d.get(v, '您输入的的键不存在！'))
```

3.31

```
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
x[:10] = sorted(x[:10])
x[10:] = sorted(x[10:], reverse=True)
print(x)
```

```
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
y1 = x[:10]
y2 = x[10:]
y1.sort()
y2.sort(reverse=True)
y1+y2
```

第三章要点是列表，元组，字典和集合各个对象的特点以及常用方法
上机实验见上机实验.ipynb 文件

第 4 章

4.1 已知 x = [1, 1]，那么执行完语句

```
for i in range(5):
    x.append(x[-1]+x[-2])
```

之后，表达式 x[-1] 的值为____13_____。

4.2 下面代码的运行结果为__1,2,_____。

```
for index, (f, s) in enumerate(zip((1,2,3), [4,5])):
    print(f, end=',')
```

4.3 下面代码的运行结果为__0,1,2,_____。

```
for i in range(3):
    print(i, end=',')
```

4.4 下面代码的运行结果为_____97_____。

```
for n in range(100, 1, -1):
    for i in range(2, n):
        if n%i == 0:
```

```
            break
        else:
            print(n)
            break
```

4.5 下面代码的运行结果为_____A_____。

```
score = 97
degree = 'DCBAAE'
index = (score - 60) // 10
if index >= 0:
    result = degree[index]
else:
    result = degree[-1]
print(result)
```

4.6 多选题：下面表达式中作为条件表达式时等价于 True 的有哪些？

A. 3+5　　B. []　　C. {3}　　D. -3

注意：条件表达式中不允许使用赋值运算符

4.7 多选题：下面表达式中作为条件表达式时等价于 True 的有哪些？

A. 3+5　　B. {}　　C. {3}　　D. 0

4.8 多选题：下面表达式中作为条件表达式时等价于 True 的有哪些？

A. 3+5　　B. [0]　　C. {3}　　D. -3

4.9 多选题：下面表达式中作为条件表达式时等价于 True 的有哪些？

A. 0　　B. {3:5}　　C. {3}　　D. 'ab'

4.10 分析逻辑运算符“or”的短路求值特性。

假设有表达式“表达式 1 or 表达式 2”，如果表达式 1 的值等价于 True，那么无论表达式 2 的值是什么，整个表达式的值总是等价于 True。此时不需要再计算表达式 2 的值，直接以表达式 1 的值作为整个表达式的值。如果表达式 1 的值等价于 False，这时无法确定整个表达式等价于 True 还是 False,所以需要继续计算表达式 2 的值并将其作为整个表达式的值。

4.11 编写程序，运行后用户输入 4 位整数作为年份，判断其是否为闰年。如果年份能被 400 整除，则为闰年；如果年份能被 4 整除但不能被 100 整除也为闰年。

```
x = input('请输入 4 位年份：')
x = int(x)
if x%400==0 or (x%4==0 and not x%100==0):
        print('是闰年')
else:
        print('不是闰年')
```

4.12 编写程序，生成一个包含 50 个随机整数的列表，然后删除其中所有奇数。（提示：从后向前删。）

```
import random
x = [random.randint(0,100) for i in range(50)]
print(x)
for i in range(len(x))[::-1]:
        if x[i]%2 == 1:
```

```
                del x[i]
print(x)
```

或者 y = [i for i in x if i%2==0]

4.13 编写程序，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。（提示：使用切片。）

```
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
x[::2] = sorted(x[::2], reverse=True)
print(x)
```

```
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
y = x[::2]
y.sort(reverse=True)
x[::2] = y
print(x)
```

4.14 编写程序，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。

```
x = int(input('请输入一个小于 1000 的整数：'))
t = x
i = 2
result = []
while True:
    if t==1:
        break
    if t%i == 0:
        result.append(i)
        t = t//i
    else:
        i += 1
print(x, '=', '*'.join(map(str,result)))
```

或者参照上课的 factors()函数 P109 页

4.15 编写程序，至少使用 2 种不同的方法计算 100 以内所有奇数的和。

```
print(sum([i for i in range(1,100) if i%2==1]))
print(sum(range(1,100)[::2]))
print(sum(range(1,100,2)))
```

4.16 编写程序，输出所有由 1、2、3、4 这四个数字组成的素数，并且在每个素数中

每个数字只使用一次。

```
import itertools
def is_prime(n):
    """检查一个数是否为素数"""
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

digits = [1, 2, 3, 4]
# 生成所有可能的排列（1 位、2 位、3 位、4 位数）
for r in range(1, 5):
    for num_tuple in itertools.permutations(digits, r):
        num = int(''.join(map(str, num_tuple)))
        if is_prime(num):
            print(num)
```

4.17 编写程序，实现分段函数计算，如下表所示。

x	y
x<0	0
0<=x<5	x
5<=x<10	3x-5
10<=x<20	0.5x-2
20<=x	0

```
x = float(input('请输入 x：'))
if 0<=x<5:
        print(x)
elif 5<=x<10:
        print(3*x-5)
elif 10<=x<20:
        print(0.5*x-2)
else:
        print(0)
```

第四章要点是程序控制结构的熟悉，包括常用的循环和选择组合语句，break 和 continue 的区别；for 和 while 循环的不同应用场合。课后提升自己程序阅读和编程能力。

第 5 章

5.1 在函数内部可以通过关键字_____global_____来定义全局变量。

5.2 如果函数中没有 return 语句或者有 return 语句但不带任何返回值，那么该函数的返回值为____None_____。

5.3 包含___yield_____语句的函数可以用来创建生成器对象。

5.4 下面代码的运行结果为_____5_____。

```
def func(para):
    para = 3

n = 5
func(n)
print(n)
```

5.5 下面代码的运行结果为_____5_____。

```
def func():
    para = 3

para = 5
func()
print(para)
```

5.6 下面代码的运行结果为_____3_____。

```
def func():
    global para
    para = 3

para = 5
func()
print(para)
```

5.7 下面代码的运行结果为_____[3]_____。

```
def func():
    para[0] = 3

para = [5]
func()
print(para)
```

5.8 下面代码的运行结果为____3_____。

```
value = 3
def func(para=value):
    print(para)

value = 5
```

```
func()
```

当你在定义函数时给参数设置默认值（比如 **`para=value`**），Python 会在 **函数定义的那一刻** 计算这个默认值，并 **固定** 它。之后即使 **`value`** 变量被修改，默认值也不会再改变.

```
value = 3
def func(para=value):
    print(para)

value = 5
func(para=value)  或者 func(5)
```

结果是_____?

5.9 下面代码的运行结果为___[3,3,3]_____。

```
def func(para=[]):
    para.append(3)
    return para

func()
func()
print(func())
```

5.10 下面代码的运行结果为_____{5:3}_____。

```
def func(a, b):
    return {b: a}

print(func(3, 5))
```

5.11 下面代码的运行结果为_____5_____。

```
from operator import eq

s1 = 'Python'
s2 = 'python'
print(sum(map(eq, s1, s2)))
```

5.12 下面代码的运行结果为_____246_____。

```
def func(a, n):
    result, each = a, a
    for _ in range(n-1):
        each = each*10 + a
        result = result + each
    return result

print(func(2, 3))
```

each = 22， result = 24

Each = 222， reslut = 246

_:在 for _ in range(n-1) 中，_ 只是一个忽略变量名的占位符，没有特殊功能。

只有在 交互式环境 中，_ 才会自动引用上一个结果。

5.13 下面代码的运行结果为____0 1_____。

```
def func():
    for i in range(10):
        if i > 3:
            return i
        yield i

r = func()
print(next(r), next(r))
```

return 在生成器中会终止迭代，不会直接输出值。

5.14 下面代码的运行结果为__0 1 2 3_____。

```
def func():
    for i in range(10):
        if i > 3:
            return i
        yield i

r = func()
print(next(r), *r)
```

```
def func():
    for i in range(10):
        if i > 3:
            yield i  # 改为 yield 而不是 return
            return   # 终止生成器
        yield i

r = func()
print(list(r))  # 输出 [0, 1, 2, 3, 4]
```

5.15 下面代码的运行结果为___0 1 2 3_____。

```
def func():
    for i in range(10):
        if i > 3:
            return i
        yield i

r = func()
print(*r)
```

5.16 假设已从标准库 functools 导入 reduce()函数，那么表达式 reduce(lambda x, y: x*y, [1, 2, 3]) 的值为____6____。

5.17 假设已成功执行语句 from functools import reduce 和 from operator import or_，那么表达式 reduce(or_, [{1},{2},{3}]) 的值为_____{1,2,3}_____。

5.18 假设已成功执行语句 from functools import reduce 和 from operator

import and_，那么表达式 reduce(and_, [{1},{2},{3}]) 的值为___set()_____。

5.19 已知函数定义

```
def func(a, b, c, *p):
    print(len(p))
```

那么语句 func(1, 2, 3, 4) 输出结果为____1_____。

5.20 下面代码的运行结果为_____3_____。

```
x = 3
def modify():
    x = 5

modify()
print(x)
```

5.21 下面代码的运行结果为_____[5]_____。

```
x = [3]
def modify():
     x[0] = 5

modify()
print(x)
```

5.22 单选题：下面代码的运行结果为？

```
x: int = [3]
print(x)
```

A. 3　　B. [3]　　C. {3}　　D. 代码错误，无法运行

5.23 单选题：下面代码的运行结果为？

```
def func():
    global x
func()
print(x)
```

A. 0　　B. []　　C. {}　　D. 出错抛出异常

5.24 单选题：已知函数定义 def func(a, b, c, *p): pass，那么通过语句 func(1, 2, 3, 4, 5) 调用函数时，在函数内部形参 p 的值为？

A. 4　　B. 5　　C. (4, 5)　　D. 出错抛出异常

5.25 多选题：关于函数的描述错误的有？

A. 只能使用关键字 def 定义函数，没有其他方式了

B. 函数属于可调用对象

C. Python 不支持嵌套定义函数

D. Python 程序必须有 main()函数作为程序执行的入口

5.26 多选题：关于函数参数的描述正确的有？

A. 函数的形参在函数内可以作为局部变量直接使用

B. 如果在函数内修改了形参变量的引用，对应实参的引用也会被修改

C. 调用函数时是把实参的引用传递给形参

D. 定义函数时不需要声明形参的类型，Python 会根据实参的值自动推断形参的类型

5.27 多选题：关于变量作用域的描述正确的有？

A．在函数中可以直接使用已定义的全局变量的值

B．在函数中使用已定义的全局变量的值必须先使用关键字 global 进行声明

C．如果在函数中有局部变量与外部的全局变量同名，会优先使用全局变量

D．在函数内试图修改已定义的全局变量的值必须先使用关键字 global 进行声明

5.28 判断对错：生成器函数的调用结果是一个确定的值。错 是一个生成器对象

5.29 判断对错：使用关键参数调用函数时，也必须记住每个参数的顺序和位置。错

5.30 判断对错：已知不同的三个函数 A、B、C，在函数 A 中调用了 B，函数 B 中又调用了 C，这种调用方式称作递归调用。错

5.31 判断对错：定义 Python 函数时，如果函数中没有 return 语句，则默认返回空值 None。对

5.32 判断对错：如果在函数中有语句 return 3，那么该函数一定会返回整数 3。错

5.33 判断对错：在函数内部没有任何声明的情况下直接为某个变量赋值，这个变量一定是函数内部的局部变量。对

5.34 判断对错：调用带有默认值参数的函数时，不能为默认值参数传递任何值，必须使用函数定义时设置的默认值。错

5.35 判断对错：lambda 表达式只能用来创建匿名函数，不能为这样的函数起名字。错

5.36 在 Python 程序中，局部变量会隐藏同名的全局变量吗？请编写代码进行验证。会

5.37 编写函数，判断一个整数是否为素数，并编写主程序调用该函数。

5.38 编写函数，接收一个字符串，分别统计大写字母、小写字母、数字、其他字符的个数，并以元组的形式返回结果。

```
def func(v):
        uppercase = lowercase = digit = other = 0
        for i in v:
                if 'A'<=i<='Z':
                        uppercase += 1
                elif 'a'<=i<='z':
                        lowercase += 1
                elif '0'<=i<='9':
                        digit += 1
                else:
                        other += 1
        return (uppercase,lowercase,digit,other)

    x = 'uppercase = lowercase = digit = other = 0'
    print(func(x))
    s.isdigit(); s.islower();s.isupper()
```

5.39 编写函数，可以接收任意多个整数并输出其中的最大值和所有整数之和。

```
def func(*v):
        print(max(v))
        print(sum(v))
```

5.40 编写函数，模拟内置函数 sum()。

```
def Sum(v):
	s = 0
	for i in v:
		s = s + i
	return s
```

5.41 编写函数，模拟内置函数 sorted()。

```
def mySorted(lst, reverse=False):
    sor = []
    temp = lst[::]
    while temp:
        if not reverse:
            a = min(temp)
            sor.append(a)
            #yield sor[-1]
            temp.remove(a)
        else:
            b = max(temp)
            sor.append(b)
            #yield sor[-1]
            temp.remove(b)
    return sor
```

5.42 编写函数，模拟内置函数 reversed()。

```
def myReversed(lst):
	for item in lst[::-1]:
		yield item
```

第 6 章

6.1 面向对象程序设计的三要素分别是__封装、_继承__和__多态_。

6.2 与运算符**对应的特殊方法名为_____pow__()_____，与运算符//对应的特殊方法名为_____floordiv__()_____。

6.3 假设 a 为类 A 的对象且包含一个私有数据成员__value，那么在类的外部通过对象 a 直接将其私有数据成员__value 的值设置为 3 的语句可以写作_a._A__value = 3_。

6.4 在 Python 中，不论类的名字是什么，构造方法都是 __init__()。

6.5 如果在设计一个类时实现了__contains__ ()方法，那么该类的对象会自动支持_____in_____关键字。

6.6 如果在设计一个类时实现了特殊方法__mul__()，那么该类的对象会自动支持运算符_____*_____。

6.7 如果在设计一个类时实现了特殊方法__eq__()，那么该类的对象会自动支持运算符_____==____。

6.8 如果在设计一个类时实现了特殊方法__gt__()，那么该类的对象会自动支持运算符_____>_____。

6.9 如果在设计一个类时实现了特殊方法__ne__()，那么该类的对象会自动支持运算符_____!=___。

6.10 如果在设计一个类时实现了特殊方法__ge__()，那么该类的对象会自动支持运算符_____>=_____。

6.11 下面代码的运行结果为____2_____。

```
class Demo:
    def __init__(self, value):
        self.__value = value

    def __add__(self, other):
        if isinstance(other, (int,float,complex)):
            return self.__value - other

t = Demo(5)
print(t + 3)
```

6.12 下面代码的运行结果为_____11_____。

```
class Demo:
    def __init__(self, x):
        self.x = x
    def __call__(self, a, b):
        return self.x*a + b

t = Demo(3)
print(t(2, 5))
```

6.13 判断对错：运行下面的代码，

```
class A(object):
    def __init__(self):
        self.__private()
```

```
        self.public()

    def __private(self):
        print('__private() method in A')

    def public(self):
        print('public() method in A')

class B(A):
    def __private(self):
        print('__private() method in B')

    def public(self):
        print('public() method in B')

B()
```

输出结果为：

```
__private() method in A
public() method in B
```

6.14 判断对错：运行下面的代码，

```
class A(object):
    def __init__(self):
        self.__private()
        self.public()

    def __private(self):
        print('__private() method in A')

    def public(self):
        print('public() method in A')

class C(A):
    def __init__(self):
        self.__private()
        self.public()

    def __private(self):
        print('__private() method in C')

    def public(self):
         print('public() method in C')
```

```
C()
```
运行结果为：
```
__private() method in C
public() method in C
```

6.15 判断对错：如果在设计一个类时实现了特殊方法__abs__()，那么该类的对象会自动支持内置函数 abs()。

6.16 判断对错：如果在设计一个类时实现了特殊方法__getitem__()，那么该类的对象会自动支持通过下标获取值。

6.17 判断对错：如果在设计一个类时实现了特殊方法__setitem__()，那么该类的对象会自动支持通过下标进行赋值。

6.18 判断对错：在实例成员方法中不可以访问属于类的数据成员。错

6.19 判断对错：在类方法中不可以访问属于实例的数据成员。对

6.20 判断对错：在静态方法中不可以访问属于实例的数据成员。对

6.21 判断对错：如果在自定义类中实现了特殊方法__call__()，那么这个类的所有对象都是可调用对象，可以像调用函数一样使用该类的对象。

6.22 判断对错：在基类中使用双下画线开始且不以双下画线结束的成员属于私有成员，无法被派生类继承，在派生类中不能直接访问。

6.23 判断对错：在 Python 中定义类时，如果某个成员名称前有 2 个下划线则表示是私有成员。错

6.24 判断对错：在类定义的外部没有任何办法可以访问对象的私有成员。错

6.25 判断对错：如果在派生类中没有定义构造方法，会自动继承基类的构造方法，使用派生类定义对象时自动调用基类的构造方法。对

6.26 判断对错：如果在派生类中定义了构造方法，使用派生类定义对象时不会自动调用基类的构造方法。对

6.27 判断对错：定义类时所有实例方法的第一个参数用来表示对象本身，在类的外部通过对象名来调用实例方法时不需要为该参数传值。对

6.28 判断对错：表达式 str.upper('abcd') == 'abcd'.upper() 的结果为 True。对

6.29 判断对错：已知 x = (i**2 for i in range(10))，那么表达式 x.__next__() 和 next(x) 的功能是等价的。对

6.30 判断对错：在面向对象程序设计中，函数和方法是完全一样的，都必须为所有参数进行传值。错

6.31 简单解释 Python 中以下画线开头和结束的变量名的含义。

在 Python 中，以下划线开头的变量名有特殊的含义，尤其是在类的定义中。用下划线作为前缀和后缀来表示类的特殊成员：

- _xxx：这样的对象叫做保护变量，不能用'from module import *'导入，只有类对象和子类对象能访问这些变量；
- __xxx__：系统定义的特殊成员名字；
- __xxx：类中的私有成员，只有类对象自己能访问，子类对象也不能访问到这个成员，但在对象外部可以通过“对象名._类名__xxx”这样的特殊方式来访问。

6.32 继承 6.3 节例 6-1 中的 Person 类创建 Student 类，添加新的方法用来设置学生专业，然后生成该类对象并显示信息。(见复习 ipynb 文件)

```
class Person(object):
    def __init__(self, name='', age=20, sex='man'):
        self.setName(name)
        self.setAge(age)
        self.setSex(sex)
    def setName(self, name):
        if not isinstance(name, str):
            print('姓名必须是字符串。')
            return
        self.__name = name
    def setAge(self, age):
        if not isinstance(age, int) or age<0:
            print('年龄必须是正整数。')
            return
        self.__age = age
    def setSex(self, sex):
        if sex not in ('man', 'woman'):
            print('性别必须是 "man" 或者 "woman"。')
            return
        self.__sex = sex
    def show(self):
        print(self.__name)
        print(self.__age)
        print(self.__sex)

class Student(Person):
    def __init__(self, name='', age=30, sex='man', major='Computer'):
        #调用基类构造方法初始化基类的私有数据成员
        super(Student, self).__init__(name, age, sex)
        self.setMajor(major) #初始化派生类的数据成员
    def setMajor(self, major):
        if not isinstance(major, str):
            print('专业必须是字符串。')
            return
        self.__major = major
    def show(self):
        super(Student, self).show()
        print(self.__major)

if __name__  == '__main__':
    zhangsan = Person('Zhang San', 19, 'man')
    zhangsan.show()
    lisi = Student('Li Si',32, 'man', 'Math')
    lisi.show()
```

6.33 设计一个三维向量类，并实现向量的加法、减法以及向量与标量的乘法和除法运算。

第 7 章

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

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

7.3 表达式 '{:c}'.format(65) == str(65) 的值为_____错_____。

7.4 表达式 len('abc\tdef'.expandtabs(8)) 的值为_____。

7.5 表达式 len('Hello world!'.ljust(20)) 的值为_ 20_____。

7.6 表达式 'a' + 'b' 的值为_____'ab'_____。

7.7 表达式 len('中国共产党'.encode('utf-8')) 的值为___15_____。

7.8 已知 text = '延安精神是中国共产党创造的一种革命精神，主要内容包括：实事求是、理论联系实际的精神，全心全意为人民服务的精神和自力更生艰苦奋斗的精神。'，那么表达式 text.count('精神') 的值为_____5_____。

7.9 表达式 len('::'.join(['a','b','c'])) 的值为___7_____。

7.10 单选题：表达式 '{:#<9d}'.format(666) 的值为？A

A. '666######'　　B. '######666'

C. '###666###'　　D. 表达式错误无法计算

7.11 单选题：表达式 '{},{}'.format(3, 5) 的值为？B

A. '3.5'　　B. '3,5'　　C. '35'　　D. '5,3'

7.12 单选题：表达式 '{1},{0}'.format(3, 5) 的值为？D

A. '1,0'　　B. '3,5'　　C. '0,1'　　D. '5,3'

7.13 单选题：表达式 'Beautiful is better than ugly.'.index('beautiful') 的值为？D

A. 0　　B. -1　　C. 1　　D. 引发异常

7.14 单选题：表达式 len('a,,b'.split(',')) 的值为？C

A. 1　　B. 2　　C. 3　　D. 4

7.15 单选题：表达式 len('a\t\t\tb'.split()) 的值为？B

A. 1　　B. 2　　C. 3　　D. 4

7.16 单选题：表达式 '1234'.upper() 的值为？A

A. '1234'　　B. '一二三四'　　C. '壹贰叁肆'　　D. 表达式错误

7.17 单选题：表达式 'abababababa'.rindex('aba') 的值为？C

A. 0　　B. 2　　C. 6　　D. -2

7.18 单选题：表达式 'abababababa'.count('aba') 的值为？B（不重叠）

A. 1　　B. 2　　C. 3　　D. 4

7.19 单选题：表达式 '{:->5.4s}'.format('abcdefg')[0] 的值为？B

A. 'a'　　B. '-'　　C. '>'　　D. 'g'

7.20 单选题：已知 table = ''.maketrans('abc', 'ABC')，那么表达式 'dongfuguo'.translate(table) 的值为？A

A. 'dong fuguo'　　B. 'Dong Fuguo'
C. 'Dong FuGuo'　　D. 'Dong fuguo'

7.21 假设有一段英文，其中有单独的字母 I 误写为 i，请编写程序进行纠正。

```
text = "If the implementatIon is hard to explain, it's a bad idea."
result = ''

for index, ch in enumerate(text):
    if ch=='I' and not text[index-1].isalpha() and not
text[index+1].isalpha():
        ch = 'i'
    result += ch

print(result)
```

```
import re
x = 'If the implementatIon is hard to explain, it is a bad idea.'
x = re.sub(r'\bI\b', r'i', x)
print(x)
```

7.22 假设有一段英文，其中有单词中间的字母 i 误写为 I，请编写程序进行纠正。

```
text = "If the implementatIon is hard to explain, it's a bad idea."
result = ''

for index, ch in enumerate(text):
    if ch=='I' and text[index-1:index].isalpha() and
text[index+1:index+2].isalpha():
        ch = 'i'
    result += ch

print(result)
```

```
import re
x = 'If the implementatIon is hard to explain, it is a bad idea.'
x = re.sub(r'\BI\B', r'i', x)
print(x)
```

第 8 章

8.1 假设已导入正则表达式模块 re，那么表达式 re.match('[a-zA-Z]+','abcDEFG000').span() 的值为_____。

8.2 假设已导入正则表达式模块 re，那么表达式 re.match('[a-z]+','abcDEFG000').span() 的值为_____。

8.3 假设已导入正则表达式模块 re，那么语句 print(re.match('[A-Z]+','abcDEFG000')) 的输出结果为_____。

8.4 假设已导入正则表达式模块 re，那么表达式 re.search('[A-Z]+','abcDEFG000').span() 的值为_____。

8.5 在设计正则表达式时，字符___? ____紧随任何其他限定符(*、+、?、{n}、{n,}、{n,m})之后时，匹配模式是“非贪心的”，匹配搜索到的、尽可能短的字符串。

8.6 假设正则表达式模块 re 已导入，那么表达式 re.sub('\d+', '1', 'a12345bbbb67c890d0e') 的值为__'a1bbbb1c1d1e'_____。

8.7 正则表达式元字符_____+___用来表示该符号前面的字符或子模式 1 次或多次出现。

8.8 正则表达式元字符_____*___用来表示该符号前面的字符或子模式 0 次或多次出现。

8.9 假设已导入模块 re，表达式 re.search(r'\w*?(?P<f>\b\w+\b)\s+(?P=f)\w*?', 'Beautiful is is better than ugly.').group(0) 的值为____'is is'_____。

8.10 假设已导入模块 re，表达式 re.search(r'(?P<f>\b\w+\b)\s+(?P=f)', 'Beautiful is is better than ugly.').group(0) 的值为__'is is'_____。

8.11 假设已导入模块 re，表达式 re.search(r'(?P<f>\b\w+\b)\s+(?P=f)', 'Beautiful is is better than ugly.').group(1) 的值为___'is'_____。

8.12 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 max(re.findall('\d+', text), key=len) 的值为__'111' _____。

8.13 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 max(re.findall('\d+', text)) 的值为___'3'_____。

8.14 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 max(re.findall('(\d+?)[a-z]', text)) 的值为___'3'_____。['111', '22', '3']

8.15 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 re.findall('(\d)[a-z]', text) 的值为___['1', '2', '3']___。

8.16 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 len(re.findall('(\w+)[a-z]', text)) 的值为___1_____。

8.17 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 len(re.split('[abc]+', text)) 的值为____4___。

8.18 已知 text = '111a22bb3ccc'，并且已导入正则表达式模块 re，那么表达式 len(re.sub('[abc]+', '', text)) 的值为___6____。

8.19 单选题：假设已导入模块 re，那么表达式 re.findall('\d{3,}', 'a12b345ccc56789') 的值为?

A. ['345']　　B. ['345', '56789']

C. ['56789']　　D. []

8.20 单选题：假设已导入模块 re，那么表达式 re.findall('\d{3}',

'a12b345ccc567890') 的值为?

A. ['345', '567', '890']　　B. ['345']

C. []　　D. ['345', '567890']

8.21 单选题：假设已导入模块 re，那么表达式 re.findall('\d{1,3}', 'a12b345ccc56789') 的值为?

A. ['12', '345']　　B. ['12', '345', '567', '89']

C. ['12', '89']　　D. ['345', '567']

8.22 单选题：假设已导入模块 re，那么表达式 re.findall('\d{,3}', 'a12b345ccc56789') 的值为?

A. ['12', '345', '567', '89']

B. ['', '12', '', '345', '', '', '', '567', '89', '']

C. ['12', '345']　　D. ['12', '89']

8.23 单选题：假设已导入模块 re，那么表达式 re.findall('abc{,3}?', 'abccc') 的值为?

A. ['ab']　　B. ['abc']　　C. ['abccc']　　D. 'ab'

8.24 单选题：假设已导入模块 re，那么表达式 re.findall('abc{3}?', 'abccc') 的值为?

A. ['ab']　　B. ['abc']　　C. ['abccc']　　D. 'ab'

8.25 单选题：已知 x = 'a234b123c'，并且 re 模块已导入，则表达式 ','.join(re.split('\d+', x)) 的值为?

A. 'a,b,c'　　B. 'a,b,c,'

C. '234,123'　　D. '2,3,4,1,2,3'

8.26 单选题：已知 x = 'a234b123c45'，并且 re 模块已导入，则表达式 ','.join(re.split('\d+', x)) 的值为?

A. 'a,b,c'　　B. 'a,b,c,'

C. '234,123'　　D. '2,3,4,1,2,3'

8.27 单选题：已知 x = 'a234bb123c45'，并且 re 模块已导入，则表达式 ','.join(re.findall('\d+', x)) 的值为?

A. '234,123,45'　　B. '234,123,45,'

C. 'a,bb,c'　　D. '23412345'

8.28 单选题：已知 x = 'a234bb123c45'，并且 re 模块已导入，则表达式 ','.join(re.findall('[a-z]+', x)) 的值为?

A. 'a,bb,c'　　B. 'a,bb,c,'　　C. 'a,b,b,c'　　D. 'a,b,b,c,'

8.29 单选题：已知 x = 'a234bb123c45'，并且 re 模块已导入，则表达式 ','.join(re.findall('[a-z]', x)) 的值为?

A. 'a,bb,c'　　B. 'a,bb,c,'　　C. 'a,b,b,c'　　D. 'a,b,b,c,'

8.30 单选题：已知 x = 'a234bb123c45'，并且 re 模块已导入，则表达式 ','.join(re.findall('[a-z]{2}', x)) 的值为?

A. 'a,bb,c'　　B. 'a,bb,c,'　　C. 'a,b,b,c'　　D. 'bb'

8.31 单选题：假设正则表达式模块 re 已正确导入，那么表达式 re.findall('\d', 'abcd1234') 的值为?

A. ['1', '2', '3', '4']　　B. ['1234']

C. ['1, 2, 3, 4']　　D. '1234'

8.32 单选题：假设正则表达式模块 re 已正确导入，那么表达式 re.findall('\d+',

'abcd1234') 的值为?

A. ['1', '2', '3', '4']　　B. ['1234']

C. ['1, 2, 3, 4']　　D. '1234'

8.33 单选题:假设正则表达式模块 re 已正确导入,那么表达式 re.findall('\d+?', 'abcd1234') 的值为?

A. ['1', '2', '3', '4']　　B. ['1234']

C. ['1, 2, 3, 4']　　D. '1234'

8.34 单选题:假设正则表达式模块 re 已正确导入,那么表达式 re.findall('\w+', 'abcd1234') 的值为?

A. ['abcd1234']　　B. 'abcd1234'　　C. 'wwwwwwww'

D. ['a', 'b', 'c', 'd', '1', '2', '3', '4']

8.35 单选题:假设正则表达式模块 re 已正确导入,那么表达式 re.sub('(.\s)\\1', '\\1', 'a a a a a bb') 的值为?

A. 'a a a bb'　　B. 'a bb'　　C. 'bb'　　D. 'a a a'

8.36 单选题:假设正则表达式模块 re 已正确导入,那么表达式 re.sub('(.\s)\\1+', '\\1','a a a a a bb') 的值为?

A. 'a a a bb'　　B. 'a bb'　　C. 'bb'　　D. 'a a a'

8.37 单选题:假设正则表达式模块 re 已正确导入,那么表达式 re.sub('(.\s)\\1+', '\101', 'a a a a a bb') 的值为?

A. 'a a a bb'　　B. 'a bb'　　C. 'Abb'　　D. 'a a a'

8.38 单选题:假设正则表达式模块 re 已导入,那么表达式 re.sub('ab|bc', '', 'abcabc') 的值为?

A. 'cc'　　B. ''　　C. 'c'　　D. 'aa'

8.39 单选题:假设正则表达式模块 re 已导入,那么表达式 re.sub('bc|ab', '', 'abcabc') 的值为?

A. 'cc'　　B. ''　　C. 'c'　　D. 'aa'

8.40 单选题:已知 example = 'Beautiful is better than ugly.',假设正则表达式 re 已导入,那么表达式 re.findall('\\bb.+\\b', example) 的值为?

A. ['better than ugly']　　B. ['better']

C. 'better than ugly'　　D. 'better'

8.41 单选题:已知 example = 'Beautiful is better than ugly.',假设正则表达式 re 已导入,那么表达式 re.findall('\\bb.+?\\b', example) 的值为?

A. ['better than ugly']　　B. ['better']

C. 'better than ugly'　　D. 'better'

8.42 单选题:已知 example = 'Beautiful is better than ugly.',假设正则表达式 re 已导入,那么表达式 len(re.findall('\\b\w.+?\\b', example)) 的值为?

A. 30　　B. 4　　C. 5　　D. 1

8.43 单选题:假设已导入正则表达式模块 re,那么表达式 len(re.findall('\d+\.\d+\.\d+', 'Python 2.7.17,Python 3.9.6'))的值为?

A. 2　　B. 3　　C. 6　　D. 1

8.44 单选题:假设已导入正则表达式模块 re,已知 s = '<html><head>This is head.</head><body>This is body.</body></html>' 和 pattern = r'<html><head>(.+)</head><body>(.+)</body></html>',那么表达式 re.findall(pattern, s) 的值为?

A. [('This is head.', 'This is body.')]
B. ['This is head.', 'This is body.']
C. ('This is head.', 'This is body.')　　D. ['This is head.']

8.45 单选题：假设已导入模块 re，有 text = '<table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td></tr></table>'，那么表达式 re.findall('<td>(.+?)</td>', text) 的值为？

A. ['1', '2', '3', '4']　　B. ['1', '2']
C. ['3', '4']　　D. ['1234']

8.46 单选题：假设已导入模块 re，有 text = '<table><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td></tr></table>'，那么表达式 re.findall('<td>(.+?)</td><td>(.+?)</td>', text) 的值为？

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

8.47 单选题：假设已导入正则表达式模块 re，那么表达式 max(re.findall('\d+', 'abcdefg'), key=len, default='no') 的值为？对象为空的默认值

A. 'n'　　B. 'o'　　C. 0　　D. 'no'"

8.48 单选题：假设已导入正则表达式模块 re，那么表达式 max(re.findall('\d+', 'abc1d22e333fg'), key=len, default='no') 的值为？

A. '33'　　B. '333'　　C. 3　　D. 333

8.49 单选题：假设已导入正则表达式模块 re，且有 example = 'ShanDong Institute of Business and Technology'，那么表达式 re.findall(r'\b\w*?g\b', example) 的值为？

A. ['ShanDong']　　B. 'ShanDong'
C. ['Technology']　　D. 'Technology'

8.50 单选题：假设已导入正则表达式模块 re，且有 example = 'ShanDong Institute of Business and Technology'，那么表达式 re.findall(r'\b\w+?g\w+?\b', example) 的值为？

A. ['ShanDong']　B. 'ShanDong'　C. ['Technology']　D. 'Technology'

8.51 单选题：假设已导入正则表达式模块 re，且有 example = 'ShanDong Institute of Business and Technology'，那么表达式 re.findall(r'\b\w*?g\w*?\b', example) 的值为？

A. ['ShanDong']　　B. ['ShanDong', 'Technology']
C. ['Technology']　　D. ['ShanDong Technology']

8.52 单选题：假设已导入正则表达式模块 re，且有 example = 'ShanDong Institute of Business and Technology'，那么表达式 re.findall(r'\bB\w+?\b', example) 的值为？

A. ['Business']　　B. 'Business'　　C. ['B']　　D. 'B'

8.53 单选题：假设已导入正则表达式模块 re，且有 example = 'Beautiful is better than ugly.'，那么表达式 re.findall(r'\bb\w+?\b', example, re.I) 的值为？

A. ['Beautiful']　　B. ['better']
C. ['Beautiful', 'better']　　D. ['Beautiful better']

8.54 单选题：假设已导入正则表达式模块 re，且有 example = r'one1two2three3four4five5555six6seven7eight88nine999ten'，那么表达式 len(re.split('\d+', example)) 的值为？

A. 9　　B. 10　　C. 54　　D. 15

8.55 单选题：假设已导入正则表达式模块 re，且有 example = r'one1two2three3four4five5555six6seven7eight88nine999ten'，那么表达式 len(re.sub(r'\d+', '', example)) 的值为？

A. 10　　B. 39　　C. 54　　D. 15

8.56 单选题：假设已导入正则表达式模块 re，那么表达式 len(re.split('\d', 'a1b23c456')) 的值为？

A. 3　　B. 4　　C. 6　　D. 7

8.57 单选题：假设已导入正则表达式模块 re，那么表达式 len(re.split('\d+', 'a1b23c456')) 的值为？

A. 3　　B. 4　　C. 6　　D. 7

8.58 判断对错：正则表达式元字符'^'一般用来表示从字符串开始处进行匹配，用在一对方括号中的时候则表示反向匹配，不匹配方括号中的字符。对

8.59 判断对错：正则表达式元字符'\s'用来匹配单个任意空白字符。对

8.60 判断对错：正则表达式元字符'\d'用来匹配单个任意数字字符。对

8.61 判断对错：正则表达式元字符'\w'可以匹配单个字母、数字或下划线。对

8.62 判断对错：正则表达式'[a-z]'可以匹配单个小写字母。对

8.63 判断对错：正则表达式'[^a-z]'可以匹配单个小写字母。错

8.64 判断对错：正则表达式'[^a-z]'可以匹配单个大写字母。错

8.65 判断对错：正则表达式'[a-z]'只能匹配小写字母 a、小写字母 z 和减号，不能匹配其他字符。错

8.66 判断对错：作为正则表达式时，字符串'\b'和 r'\b'的含义一样。错

8.67 判断对错：作为正则表达式时，字符串'\1'和 r'\1'的含义一样。错

8.68 判断对错：作为正则表达式时，字符串'\n'和 r'\n'的含义一样。对

8.69 判断对错：作为正则表达式时，字符串'\ab'和 r'\ab'的含义一样。错

8.70 判断对错：假设 re 模块已成功导入，并且有 pattern = re.compile('^'+'\.'.join([r'\d{1,3}' for i in range(4)])+'$')，那么表达式 pattern.match('192.168.1.103') 的值为 None。错

8.71 判断对错：正则表达式'^http'只能匹配所有以'http'开头的字符串。对

8.72 编写程序，定义函数 main()接收一个任意字符串 text 作为参数，要求删除其中除下画线之外的所有中英文标点符号，返回处理后的新字符串。例如，main('富强、民主、文明、和谐；自由、平等、公正、法治；爱国、敬业、诚信、友善。')返回'富强民主文明和谐自由平等公正法治爱国敬业诚信友善'。

```
import re

def main(s):
    return re.sub('[^\w ]', '', s)

print(main('富强、民主、文明、和谐。自由、平等、公正、法治。爱国、敬业、诚信、友善。'))
```

8.73 有一段英文文本，其中有个单词连续重复了 2 次，编写程序检查重复的单词并只保留一个。

```
import re
```

```
x = 'This is is a desk.'
x = re.sub(r'(\w+)\s\1', r'\1', x)
print(x)
```

8.74 编写程序，用户输入一段英文，然后输出这段英文中所有长度为 3 个字母的单词。

```
import re

x = input('Please input a string:')
y = re.findall(r'\b[a-zA-Z]{3}\b', x)
print(y)
```

第 9 章

9.1 按数据组织形式，可以把文件分为___文本文件___和二进制文件两大类。

9.2 Python 内置函数_____open()___用来打开或创建文件并返回文件对象。

9.3 Python 内置函数 open()的参数_____mode_____用来指定文件打开模式。

9.4 Python 内置函数 open()的参数_____encoding_____用来指定打开文本文件时所使用的编码格式。

9.5 使用上下文管理关键字___with____可以自动管理文件对象，不论何种原因结束该关键字中的语句块，都能保证文件被正确关闭并且已写入的内容确实保持到硬盘上。

9.6 对于文本文件，使用 Python 内置函数 open()以读文本模式成功打开后返回的文件对象___可以_____（可以、不可以？）使用 for 循环直接迭代。

9.7 已知当前文件夹中有纯英文文本文件 readme.txt，请填空完成功能把 readme.txt 文件中的所有内容复制到 dst.txt 中，with open('readme.txt') as src, open('dst.txt', __'w'__) as dst:dst.write(src.read())。

9.8 Python 扩展库_____支持 Excel 2007 或更高版本文件的读写操作，是目前使用较多的扩展库，除此之外还有 xlwings 也是不错的扩展库。

9.9 使用 Python 读写 Word 2007 文件，需要安装_____扩展库，这是使用较多的一个扩展库，除此之外还有 docx2python 也是不错的扩展库。

9.10 判断对错：使用内置函数 open()以二进制模块打开文件时，也可以使用参数 encoding 指定编码格式。错

9.11 判断对错：使用内置函数 open()的 'r' 模式打开包含多行内容的文本文件并返回文件对象 fp，那么表达式 fp.readline()[-1] 的值一定为'\n'。对

9.12 判断对错：使用内置函数 open()的 'r' 模式打开包含多行内容的文本文件并返回文件对象 fp，那么表达式 fp.readlines()[0][-1] 的值为'\n'。对

9.13 判断对错：使用内置函数 open()打开文本文件时，参数 encoding 不重要，直接使用默认值就可以。错

9.14 判断对错：假设当前文件夹中包含非空文件 test.dat，那么先后执行语句 fp = open('test.dat', 'rb')、print(fp.read(5))、fp.seek(0)、print(fp.read(5))，连续两次输出的内容是一样的。对

9.15 判断对错：使用内置函数 open()且以'w'模式打开的文件，文件指针默认指向文件尾。错

9.16 判断对错：使用内置函数 open()打开文本文件时，不能指定'rb'，只能使用'r'模式和恰当的 encoding 参数。错

9.17 判断对错：使用内置函数 open()且以'a'模式打开的文件，文件指针默认指向文件尾。对

9.18 判断对错：使用内置函数 open()且以'ab'模式打开的文件，文件指针默认指向文件尾。对

9.19 判断对错：使用内置函数 open()打开文件时，只要文件路径正确就总是可以正确打开的。错

9.20 判断对错：二进制文件不能使用记事本程序打开，会报错。错

9.21 判断对错：运行下面的代码，

```
fp = open('text.txt', 'w', encoding='utf8')
for i in range(10):
    fp.write(str(i//(i-3)))
fp.close()
```

错

代码会抛出异常，但是 text.txt 文件中会写入一部分内容。

9.22 判断对错：运行下面的代码，

```
with open('text.txt', 'w', encoding='utf8') as fp:
    for i in range(10):
        fp.write(str(i//(i-3)))
```

代码会抛出异常，但是 text.txt 文件中会写入一部分内容。对

9.23 判断对错：内置函数 open()使用'w'模式打开的文件，不仅可以往文件中写入内容，也可以从文件中读取内容。错

9.24 判断对错：内置函数 open()使用'r'模式打开的文件，只能读取其中的内容，不能写入任何新内容。对

9.25 判断对错：内置函数 open()使用'r+'模式打开的文件，只能读取其中的内容，不能写入任何新内容。错

9.26 判断对错：读写文件时，只要程序中调用了文件对象的 close()方法，就一定可以保证文件被正确关闭。错

9.27 判断对错：使用扩展库 openpyxl 的函数 Workbook()创建新工作簿时，默认情况下是完全空白的，里面没有工作表，必须自己使用工作簿对象的 create_sheet()方法创建工作表才能写入数据。错

9.28 判断对错：内置函数 open()以'r'模式打开的文本文件对象是可遍历的，可以使用 for 循环遍历文件中每行文本。对

9.29 判断对错：以读模式打开文件时，文件指针指向文件开始处。对

9.30 判断对错：以追加模式打开文件时，文件指针指向文件尾。对

9.31 判断对错：CSV 格式的文件属于文本文件。

9.32 判断对错：Python 源程序文件属于二进制文件。

9.33 判断对错：视频文件属于二进制文件。

9.34 判断对错：音频文件属于二进制文件。

9.35 判断对错：扩展命为.py 和.pyw 的 Python 程序文件属于文本文件。对

9.36 判断对错：扩展名为.py 的文件属于文本文件，扩展名为.pyw 的文件属于二进制文件。

9.37 判断对错：扩展名为.whl 的文件属于二进制文件。

9.38 判断对错：扩展名为.pyd 的文件属于二进制文件。

9.39 判断对错：Python 的主程序文件 python.exe 属于二进制文件。

9.40 判断对错：Python 程序进行伪编译后得到的.pyc 文件属于二进制文件。

9.41 判断对错：对字符串信息进行编码以后，必须使用同样的或者兼容的编码格式进行解码才能还原本来的信息。

9.42 假设有一个英文文本文件 example.txt，编写程序读取其内容，并将其中的大写字母变为小写字母，小写字母变为大写字母，输出原始内容和处理后的内容。

```
with open('example.txt', 'r') as src:
    for line in src:
        print(line)
        print(line.swapcase())
```

9.43 编写程序，使用 pickle 模块将包含学生成绩的字典保存为二进制文件，然后读取内容并显示。

```
import pickle

score = {'张三':98, '李四':90, '王五':100}

with open('score.dat', 'wb') as fp:
    pickle.dump(score, fp)

with open('score.dat', 'rb') as fp:
    result = pickle.load(fp)

print(result)
```

9.44 简单解释文本文件与二进制文件的区别。

第 10 章

10.1 Python 标准库 os 中用来列出指定文件夹中的文件和子文件夹并返回列表的函数是___listdir()_____。

10.2 Python 标准库 os 中用来创建文件夹的函数是___mkdir()__，如果要创建的文件夹已存在，会报错抛出异常。

10.3 Python 标准库 os.path 中用来判断指定文件是否存在的函数是_____。

10.4 Python 标准库 os.path 中用来判断指定路径是否为文件的函数是_____。

10.5 Python 标准库 os.path 中用来判断指定路径是否为文件夹的函数是_____。

10.6 Python 标准库 os.path 中用来分割指定路径中的文件扩展名的函数是

___splitext()_____。

10.7 标准库 os 中的函数___remove() __用来删除指定的文件，如果文件具有只读属性或当前用户不具有删除权限则无法删除并引发异常。

10.8 标准库 os 中的函数_____用来启动相应的外部程序并打开参数路径指定的文件，如果参数为网址 URL 则打开默认的浏览器程序。

10.9 标准库 os.path 中的函数_____用来获取参数指定的文件的大小，单位为字节。

10.10 标准库 os.path 中的函数_____用来获取参数指定的文件的最后修改时间。

10.11 标准库 os.path 中的函数_____join_____用来把多个路径连接成为一个完整的路径，并插入适当的路径分隔符（在 Windows 操作系统中为反斜线）。

10.12 标准库 os.path 中的函数___basename()_用来获取参数指定的路径中最后一个组成部分（通常为文件名），例如如果把路径 r'C:\Windows\notepad.exe'作为参数传递给该函数则返回字符串'notepad.exe'。

10.13 标准库 os.path 中的函数_dirname()__用来获取参数指定的路径中最后一个路径分隔符前面的部分（通常为文件夹名），例如如果把路径 r'C:\Windows\notepad.exe'作为参数传递给该函数则返回字符串'C:\\Windows'。

10.14 标准库 shutil 中的函数_____可以用来创建 tar 或 zip 格式的压缩文件。

10.15 标准库 shutil 中的函数_____可以用来解压缩 tar 或 zip 格式的压缩文件。

10.16 假设已执行语句 from os.path import splitext 导入对象，那么表达式 splitext(r'C:\Python39\python.exe')[1] 的值为___'.exe'_____。

10.17 假设已执行语句 from os.path import split 导入对象，那么表达式 split(r'C:\Python39\python.exe')[1] 的值为__'python.exe'____。

10.18 判断对错：假设已导入标准库函数 os.listdir()和 os.path.exists()，且已知 fns = listdir(r'C:\Windows')，其中 C:\Windows 为安装操作系统的非空文件夹，那么表达式 exists(fns[0]) 的值一定为 True。

10.19 判断对错：标准库 os 中的函数 remove()可以删除带有只读属性的文件。

10.20 判断对错：已知当前工作目录为 C:\python39，那么下面代码的运行结果是若干个 True。

```
from os import listdir
from os.path import exists

for fn in listdir(r'D:\\'):
    print(exists(fn))
```

10.21 判断对错：下面的代码运行一次和连续运行多次的结果是一样的。

```
from os import mkdir

mkdir('test')
```

错

10.22 判断对错：Python 标准库 os 中的函数 startfile()可以启动任何已关联应用程序的文件，并自动调用关联的应用程序。

10.23 判断对错：假设 os 模块已导入，那么列表推导式 [filename for filename in os.listdir('C:\\Windows') if filename.endswith('.exe')] 的作用是列出

C:\Windows 文件夹中所有扩展名为.exe 的文件。对

10.24 判断对错：Python 标准库 os.path 中的函数 isfile()可以用来测试给定的路径是否为文件。

10.25 判断对错：Python 标准库 os.path 中的函数 exists()可以用来测试给定路径的文件或文件夹是否存在。

10.26 判断对错：Python 标准库 os.path 中的函数 isdir()可以用来测试给定的路径是否为文件夹。

10.27 判断对错：Python 标准库 os 中的函数 listdir()返回包含指定路径中所有文件和文件夹名称的列表。对

10.28 判断对错：标准库 os 的 rename()函数可以实现文件移动操作，但不能跨越磁盘分区。对

10.29 判断对错:标准库 os 的 listdir()函数只能列出指定文件夹中第一层级的文件和文件夹列表，不能列出其子文件夹中的文件。对

10.31 编写代码，将当前工作目录修改为 C:\，并验证，最后将当前工作目录恢复为原来的文件夹。

10.32 编写程序，用户输入一个文件夹和一个文件名，搜索该文件夹及其子文件夹中是否存在该文件。

第 11 章

11.1 除了代码出错时会抛出异常，还可以使用_____raise_____语句主动抛出异常。

11.2 带有 else 的异常处理结构，如果 try 中的代码抛出了异常，那么 else 中的代码将_____（会、不会？）执行。

11.3 在 try...except...异常处理结构中，__except____子句用于尝试捕捉可能出现的异常。

11.4 在异常处理结构中，如果不管是否发生异常都要执行某段代码，那么应该把这段代码放在异常处理结构的___finally_____子句中。

11.5 下面代码的运行结果为_____4_____。

```
def func():
    for i in range(10):
        if i > 3:
            return i
        yield i

r = func()
for _ in range(10):
    try:
        next(r)
    except StopIteration as e:
        print(e.value)        #生成器对象的返回值
        break
```

11.6 语句 assert 3==3 _____（会、不会？）引发异常。

11.7 表达式 3/0 _____（会、不会？）引发异常。

11.8 表达式 'a' + 1 _____（会、不会？）引发异常。

11.9 判断对错：试图计算表达式 1/0 时会抛出 ZeroDivisionError 类型的异常。对

11.10 判断对错：试图计算表达式 '2' + 1 时会抛出 TypeError 类型的异常。对

11.11 判断对错：试图计算表达式 int('3.14') 时会抛出 ValueError 类型的异常。对

11.12 判断对错：使用内置函数 open()打开文件时，如果指定的文件路径错误，代码会抛出 FileNotFoundError 类型的异常。对

11.13 判断对错：试图计算表达式 'Python_xiaowu'.encode().decode('gbk') 时会抛出 UnicodeDecodeError 异常并提示无法解码。错

11.14 判断对错：已知 x = []，那么试图执行语句 x.pop() 时会抛出 IndexError 类型的异常。对

11.15 判断对错：已知 x = {}，那么试图执行语句 print(x['a']) 时会抛出 KeyError 类型的异常。对

11.16 判断对错：在 try...except...else 结构中，如果 try 块的语句引发了异常则会执行 else 块中的代码。错

11.17 判断对错：一般不建议在 try 中放太多代码，应该只包含可能会引发异常的代码。对

11.18 判断对错：一旦代码抛出异常并且没有得到正确的处理，整个程序会崩溃，并且不会继续执行后面的代码。对

11.19 判断对错：在异常处理结构中，每个 except 子句只能捕捉和处理一种类型的异常，无法同时捕捉和处理多种不同类型的异常。错

11.20 判断对错：一般不建议直接使用空的 except: 子句来捕捉异常，因为这样会捕捉绝大多数类型的异常，难以发现真正的问题。对

11.21 判断对错：使用异常处理结构时，每个 try 子句只能带一个 except 子句，不能有多个 except 子句。错

11.22 判断对错：Python 中的异常处理结构必须带有 finally 子句。错

11.23 判断对错：Python 中的异常处理结构可以不带 else 子句。对

11.24 判断对错：异常处理结构中的 finally 块中代码仍然有可能出错从而再次引发异常。对

11.25 判断对错：带有 else 子句的异常处理结构，如果 try 块中的代码不发生异常则执行 else 子句中的代码。对

11.26 判断对错：带有 else 子句的异常处理结构，如果 try 块中的代码发生异常就继续执行 else 块中的代码。错

11.27 判断对错：异常处理结构也不是万能的，用来处理异常的代码也有引发异常的可能。对

11.28 判断对错：在异常处理结构中，不论是否发生异常，finally 子句中的代码总是会执行的。对

11.29 判断对错：由于异常处理结构 try...except...finally...中 finally 里的语句块总是被执行的，所以把关闭文件的代码放到 finally 块里肯定是万无一失，一定能保证文件被正确关闭并且不会引发任何异常。错

11.30 判断对错：assert 断言语句执行时，如果要求的条件是成立的，直接执行后面的代码，和什么也没发生一样。对

11.31 判断对错：为了防止代码出现异常导致崩溃，最好把整个程序的所有代码都放入一个大的 try 块中，这样就安全了。错

11.32 判断对错：下面的代码虽然没有使用异常处理结构，但是也能完美避免输入不是整数时抛出异常，并且不影响正常输入整数时代码的功能。对

```
num = input('请输入一个整数：')
if num.isdigit():
    print(int(num))
else:
    print('输入的不是整数。')
```

11.33 单选题：表达式 3/0 会抛出下面哪种异常？

A. ZeroDivisionError　　B. TypeError

C. SyntaxError　　D. NameError

11.34 单选题：表达式 sum(1, 2, 3) 会抛出下面哪种异常？

A. ZeroDivisionError　　B. TypeError

C. SyntaxError　　D. NameError

11.35 单选题：语句 data = {[1], [2]} 会抛出下面哪种异常？

A. ZeroDivisionError　　B. TypeError

C. SyntaxError　　D. NameError

11.36 单选题：语句 data = {'a':97, 'b':98, 99, 100} 会抛出下面哪种异常？

A. ZeroDivisionError　　B. TypeError

C. SyntaxError　　D. NameError

11.37 单选题：使用语句 print(age) 试图访问一个不存在的变量 age 时会抛出下面哪种异常？

A. ZeroDivisionError　　B. TypeError

C. SyntaxError　　D. NameError

11.38 单选题：执行语句 number = int(input('请输入一个正整数：'))，输入 3.14 时会抛出下面哪种异常？

A. TypeError　B. SyntaxError　C. ValueError　D. AttributeError

11.39 单选题：已知 data 是一个非空列表对象，那么表达式 data.rindex(3) 会抛出下面哪种异常？

A. TypeError　B. SyntaxError　C. ValueError　D. AttributeError

11.40 单选题：表达式 'A' + 32 会抛出下面哪种异常？

A. TypeError　B. SyntaxError　C. ValueError　D. AttributeError

11.41 单选题：表达式 print(3(4+5)) 会抛出下面哪种异常？

A. TypeError　B. SyntaxError　C. ValueError　D. AttributeError

11.42 单选题：语句 print('Hello world) 会抛出下面哪种异常？

A. TypeError　B. SyntaxError　C. ValueError　D. AttributeError

11.43 单选题：语句 x = 3 + 5\ - 2 会抛出下面哪种异常？

A. TypeError　B. SyntaxError　C. ValueError　D. AttributeError