

第 1 章习题

一、单选题

- Python 语言属于_____。
A. 机器语言 B. 汇编语言 **C. 高级语言** D. 以上都不是
- 下列选项中,不属于 Python 特点的是_____。
A. 面向对象 **B. 运行效率高** → 低级语言
C. 可移植性 D. 免费和开源
- 下列选项中,_____实现是最常用的 Python 版本,也称之为 ClassicPython。
A. CPython B. Jython C. IronPython **D. PyPy**
- Python 内置的集成开发工具是_____。
A. PythonWin B. Pydev C. IDE **D. IDLE**
- Python 解释器的提示符为_____。
A. > B. >> **C. >>>** D. #
- Python 解释器环境中,用于表示上一次运算结果的特殊变量为_____。
A. : **B. _** C. > D. #
- _____是 Python 官方的扩展库索引,所有人都可以下载第三方库或上传自己开发的库到其中。
A. PyPI B. PyPy C. Pydev D. pip
- 以下程序设计语言中,Prolog 属于_____的编程语言。
A. 面向过程 B. 面向对象 C. 函数式 **D. 逻辑式**
- 以下程序设计语言中,Java、C#、C++ 等属于_____的编程语言。
A. 面向过程 **B. 面向对象** C. 函数式 D. 逻辑式
- 以下程序设计语言中,C 语言属于_____的编程语言。
A. 面向过程 B. 面向对象 C. 函数式 D. 逻辑式
- 以下程序设计语言中,_____属于脚本语言。
A. Python B. Java C. C++ D. C#
- 在 Python 解释器中交互式执行 Python 代码的过程一般称之为_____。
A. FIFO B. REPT C. IPO **D. REPL**
- 在 Python 解释器中同时运行多个以逗号分隔的表达式,返回结果为_____。
A. 列表 **B. 元组**
C. 逗号分隔的结果 D. 空格分隔的结果
- 官方将 PEP(Python Enhancement Proposals)分成除_____以外的三类。
A. 指导类 B. 信息类 C. 流程类 D. 标准类
- 在 Python 解释器中输入_____命令,可以输出“The Zen of Python(Python 之禅)”的内容。
A. import it B. import zen
C. print zen **D. import this**

姓名：

班级：

学号：

年 月 日

二、填空题

1. Python 语言是一种解释型、面向对象的计算机程序设计语言。
2. 用户编写的 Python 程序(避免使用依赖于系统的特性)无须修改就可以在任何支持 Python 的平台上运行,这是 Python 的可移植性好特性。
3. 在 Python 3.4 以后的版本中,pip 库用于安装管理 Python 扩展包,setuptools 库用于发布 Python 包。
4. 用户可以使用 quit() 命令或者快捷键 Ctrl + Z + enter 关闭 Python 解释器。
5. 在 Python 内置集成开发环境 IDLE 中,可以使用快捷键 F5,运行当前打开的源代码程序。
6. Python 注释以符号 # 开始,到行尾结束。
7. 在 Python 程序中导入 sys 模块后,可以通过列表 sys.argv 访问命令行参数。argv[0] 表示 Python 脚本名;argv[1] 表示第一个参数。
8. 在 Python 解释器中使用 help 函数可以进入帮助系统;输入 quit 命令,可以退出帮助系统。
9. Python 程序打包和发布最常用的是第三方的扩展包 PyInstaller,它是用于将 Python 源程序生成直接运行的程序。
10. 程序设计语言分为低级语言和高级语言两类,其中,机器语言和汇编语言属于 低级,Python 属于 高级。
11. 结构化程序设计通常采用 自顶向下、逐步求精 的程序设计方法。

姓名： 班级： 学号： 年 月 日

12. 计算思维的本质是设计和构造。

13. 将源文件转换成机器语言一般有以下两种转换方法,即编译和解释。

14. 高级程序设计语言根据执行机制不同可以分成静态语言和脚本语言两类。采用编译方式执行的语言属于静态,采用解释方式执行的语言属于脚本。

15. 高级程序设计语言根据执行机制不同可以分成静态语言和脚本语言两类。C、C++、C#、Java 等程序设计语言属于静态,Python 程序设计语言属于脚本。

16. 在 Python 中利用 pip 命令查看 pip 常用帮助信息的选项是pip -h。

17. 开发和运行 Python 程序一般包括交互式和脚本式两种方式。

18. 在 Python 解释器中输入import this命令,可以输出“The Zen of Python(Python 之禅)”的内容。

19. 人类在认识世界和改造世界过程中形成了逻辑、实践和计算三种基本的思维。

20. 在 Python 编程规范中,对于缩进与换行,每一级缩进使用4个空格。

三、思考题

1. 简述 Python 语言的主要特点。

python是一种解释型,面向对象的编程语言

简单,高级,面向对象,可扩展性好

免费,开源,可移植性好,丰富的库,可嵌入性

姓名：

班级：

学号：

年 月 日

2. 简述 Python 语言的应用范围。

- ① 科学计算
- ② 机器学习和人工智能
- ③ 网络爬虫和 Web 信息提取
- ④ Web 应用
- ⑤ 图形用户界面开发

- ① 操作系统管理
- ② 游戏开发

3. 简述 Python 2 和 Python 3 的主要区别。

Python 3 不支持向下兼容

Python 2和Python 3是Python编程语言的两个主要版本，它们之间有以下一些主要区别：

语法方面

- print函数：Python 2中，print是语句，如print "Hello"；Python 3中，print是函数，必须写成print("Hello")。
- 除法运算：Python 2中，整数相除结果为整数，如5 / 2结果是2；Python 3中，5 / 2结果是2.5，若要取整需用//运算符。
- 编码声明：Python 2默认使用ASCII编码，Python 3默认使用UTF-8编码。

数据类型方面

- 字符串类型：Python 2有str和unicode两种字符串类型，Python 3中只有str表示Unicode字符串，byte类型用于处理二进制数据。
- 整数类型：Python 2有int和long类型，Python 3统一为int类型，支持任意大小的整数。

函数和模块方面

- range函数：Python 2中range函数返回列表，xrange函数返回生成器；Python 3中只有range函数，返回的是可迭代的对象，类似Python 2中的xrange。
- 模块导入：Python 3在模块导入机制上更严格和规范，一些在Python 2中隐式的相对导入在Python 3中不被允许，需显式指定。

异常处理方面

- 语法形式：Python 2中捕获异常可以写成except Exception, e；Python 3中必须写成except Exception as e。

其他方面

- 内置函数：Python 3对一些内置函数进行了优化和调整，如zip、map、filter等函数在Python 3中返回的是迭代器，而在Python 2中返回列表。
- 兼容性：Python 2的一些第三方库没有及时更新，与Python 3存在兼容性问题。

4. Python 语言包括哪些实现？

Python 2 或 Python 3 : 语法规则

Python 的 解释器

5. Python 语言主要包括哪些集成开发环境？

- IDLE
- Jupyter Notebook
- vscode
- VS Studio
- PyCharm

6. 简述下载和安装 Python 的主要步骤。

- 下载 {
- (1) 打开 Python 官网
 - (2) 下载 Python 安装程序
- 安装 {
- (1) 运行安装程序
 - (2) 设定安装选项
 - (3) 安装程序

姓名：

班级：

学号：

年

月

日

7. 请问如何安装和管理 Python 扩展包？

Pip 指令

pip install

pip uninstall

8. 什么是 Python 解释器？如何使用 Python 解释器交互式测试 Python 代码？

Python解释器是一种软件工具，其作用是将人类编写的Python代码翻译成计算机能够理解和执行的机器语言指令，从而实现程序的运行。

运行解释器

在提示符下输入语句， print("你好")

输入 quit() 命令退出

9. 在 Python 解释器环境中的特殊变量“_”表示什么含义？

输出上一次正确的计算结果

10. 什么是 Python 源代码程序？如何运行 Python 源代码程序？

Python源代码程序是指用Python编程语言编写的、未经编译的原始代码集合，是开发者为实现特定功能或解决特定问题而创作的文本形式的程序。

python + 地址 + 源代码程序名
(hello.py)

11. 如何使用文本编辑器和命令行编写和执行 Python 源文件程序？

编写：在 IDE 中，

在 "C:\pythonpath\ch01" 下

创建 hello.py

输入 print("Hello, World!")

保存

运行， python C:\pythonpath\ch01\hello.py

姓名：

班级：

学号：

年 月 日

12. 如何使用 Python 内置集成开发环境 IDLE 编写和运行 Python 源文件程序？

打开-新建 → 编写-保存-运行-退出
ctrl+N ctrl-S F5 quit

13. 如何使用 Python 交互式帮助系统获取相关资源？

help()

P15. 例 1.24

14. 如何使用 Python 文档以获取 Python 语言及标准模块的详细参考信息？

P16. 例 1.26

15. 什么是 PEP？官方将 PEP 分成哪三类？通过阅读 PEP 可以帮助读者深入了解 Python 的哪些内容？ → 特性 & 发展历史

Python 增强提案

{ 信息类
 流程类
 标准类

第 2 章习题

一、单选题

1. 在 Python 中,合法的标识符是_____。
 A. _ B. 3C C. it's D. str
2. 在 Python 表达式中可以使用_____控制运算的优先顺序。
 A. 圆括号() B. 方括号[] C. 花括号{} D. 尖括号<>
3. 在下列 Python 语句中非法的是_____。
 A. x=y=1 B. x=(y=1)
 C. x,y=y,x D. x=1; y=1
4. 以下 Python 注释代码,不正确的是_____。
 A. # Python 注释代码 B. #Python 注释代码 1 #Python 注释代码 2
 C. """ Python 文档注释""" D. // Python 注释代码
5. 数学关系式 $2 < x \leq 10$ 表示成正确的 Python 表达式为_____。
 A. $2 < x \parallel x \leq 10$ B. $2 < x \leq 10$
 C. $2 < x \&\& x \leq 10$ D. $x > 2 \text{ or } x \leq 10$
6. 在 Python 中,正确的赋值语句为_____。
 A. $x+y=10$ B. $x=2y$ C. $x=y=30$ D. $3y=x+1$
7. 为了给整型变量 x、y、z 赋初值 10,下面正确的 Python 赋值语句是_____。
 A. $xyz=10$ B. $x=10 \ y=10 \ z=10$
 C. $x=y=z=10$ D. $x=10,y=10,z=10$
8. 为了给整型变量 x、y、z 赋初值 5,下面正确的 Python 赋值语句是_____。
 A. $x=5; y=5; z=5$ B. $xyz=5$
 C. $x,y,z=5$ D. $x=5,y=5,z=5$
9. 已知 $x=2$ 并且 $y=3$,复合赋值语句 $x * = y + 5$ 执行后,x 变量中的值是_____。
 A. 11 B. 16 C. 13 D. 26
10. 整型变量 x 中存放了一个两位数,要将这个两位数的个位数字和十位数字交换位置,例如,13 变成 31,正确的 Python 表达式是_____。
 A. $(x \% 10) * 10 + x // 10$ B. $(x \% 10) // 10 + x // 10$
 C. $(x / 10) \% 10 + x // 10$ D. $(x \% 10) * 10 + x \% 10$
11. 与数学表达式 $\frac{cd}{2ab}$ 对应的 Python 表达式中,不正确的是_____。
 A. $c * d / (2 * a * b)$ ✓ B. $c / 2 * d / a / b$ ✓
 C. $c * d / 2 * a * b$ D. $c * d / 2 / a / b$ ✓
12. 关系式 ' $a \leq aChar \leq z$ ' 表示成正确的 Python 表达式为_____。
 A. ' $a \leq aChar \&\& aChar \leq z$ ' B. ' $a \leq aChar \& aChar \leq z$ '
 C. ' $a \leq aChar \text{ and } aChar \leq z$ ' D. $aChar \geq 'a' \text{ or } aChar \leq 'z'$

姓名：

班级：

学号：

年 月 日

二、填空题

1. Python 语句分为 简单 语句和复合语句。

2. Python 使用 缩进 格式划分语句块。

3. 在 Python 中如果语句太长,可以使用 \ 作为续行符。

4. 在 Python 中在一行书写两条语句时,语句之间可以使用 分号 作为分隔符。

5. Python 使用符号 # 标示注释。

6. 在 Python 中要表示一个空的代码块,可以使用空语句 pass。

7. 计算 $2^{32}-1$ 的 Python 表达式可以书写为 2**32-1。

8. Python 表达式 $4.5/2$ 、 $4.5//2$ 和 $4.5\%2$ 的值分别为 2.25、2.0、0.5。

9. Python 表达式 $12/4-2+5*8/4\%5/2$ 的值为 1。
从左向右算

10. Python 大部分对象均为不可变对象,例如 int, float 等;而 list 等则为可变对象。

11. Python 提供了两个对象身份比较运算符 is 和 is not 来测试两个变量是否指向同一个对象;通过内置函数 type() 来测试对象的类型;通过 == 运算符判断两个变量指向的对象的值是否相同。

12. Python 语句序列“`a,b=3,4; a,b = b,a; print(a,b)`”的执行结果是 4,3。

三、思考题

1. Python 语句的主要作用是什么？Python 主要包含哪些语句？

- ① 语句是 Python 程序的过程性逻辑块，用于创建对象、变量赋值等。
- ② 赋值，条件，循环，函数定义，import

2. 在 Python 中 pass 语句的作用是什么？

表示空的代码块。

3. 在 Python 中 type(1) 的含义是什么？

获取整数 1 的数据类型

4. 在 Python 中有哪几种注释方式？

- 1. # 单行
- 2. """ 注释 """ 三个单引号

5. Python 语句的主要书写规则是什么？

- 1. 缩进规则
- 2. 语句分隔
- 3. 换行规则
- 4. 代码块结构
- 5. 命名规范
- 6. 注释规则

6. Python 表达式遵循哪些主要的书写规则？

- (1) 从左到右在同一个基准上书写
- (2) 乘号不能省略
- (3) 括号必须成对出现，只能使用圆括号
- 圆括号可以嵌套使用。

姓名:

班级:

学号:

年

月

日

7. 在 Python 中假设有 $a=10$, 写出下面表达式运算后 a 的值:
- (1) $a += a$ 20 (2) $a -= 2$ 8 (3) $a *= 2 + 3$ 50
 (4) $a /= 2 + 3$ 2 (5) $a \% = a - a \% 4$ 2 (6) $a //= a - 3$ 1

8. 当运行测试输入 6789 时, 写出下面 Python 程序的执行结果。

```
num = int(input("请输入一个整数:"))
while (num != 0):
    print(num % 10, end = '')
    num = num // 10
```

结果: 9 8 7 6

9. 下列 Python 语句的输出结果是 <class .function >

```
def f(): pass
print(type(f()))
```

10. 下列 Python 语句的输出结果是 True, True
False, True, True

```
x = y = [1, 2]; x.append(3)
print(x != y, x != y, end = '')
z = [1, 2, 3]
print(x == z, x == z, y == z)
```

11. 阅读下面 Python 程序, 请问绘制的图形是什么?

```
import turtle as t
for i in range(1,5):
    t.fd(50)
    t.left(90)
t.h
```

正方形



第 3 章习题

一、单选题

A

1. 下面 Python 循环体执行的次数与其他不同的是_____。

A. $i = 0$ 11次 + 1次否

B. $i = 10$ 10次 + 1次否

while($i \leq 10$):

while($i > 0$):

print(i)

print(i)

$i = i + 1$

$i = i - 1$

C. for i in range(10): 1-10

D. for i in range($10, 0, -1$):

print(i) 10次

print(i) 10次

A

2. 执行下列 Python 语句将产生的结果是_____。

$x = 2; y = 2.0$

if($x == y$): print("Equal")

else: print("Not Equal")

A. Equal

B. Not Equal

C. 编译错误

D. 运行时错误

B

3. 执行下列 Python 语句将产生的结果是_____。

$i = 1$

if (i): print (True)

else: print (False)

A. 输出 1

B. 输出 True

C. 输出 False

D. 编译错误

A

4. 用 if 语句表示如下分段函数 $f(x)$, 下面程序不正确的是_____。

$$f(x) = \begin{cases} 2x+1 & x \geq 1 \\ 3x/(x-1) & x < 1 \end{cases}$$

A. if($x \geq 1$): $f = 2 * x + 1$

B. if ($x \geq 1$): $f = 2 * x + 1$

$f = 3 * x / (x - 1)$

if ($x < 1$): $f = 3 * x / (x - 1)$

C. $f = 2 * x + 1$

D. if ($x < 1$): $f = 3 * x / (x - 1)$

if ($x < 1$): $f = 3 * x / (x - 1)$

else: $f = 2 * x + 1$

B

5. 下面的 if 语句统计满足“性别(gender)为男、职称(rank)为教授、年龄(age)小于 40 岁”条件的人数, 正确的语句为_____。

A. if (gender == "男" or age < 40 and rank == "教授"): $n += 1$

B. if (gender == "男" and age < 40 and rank == "教授"): $n += 1$

C. if (gender == "男" and age < 40 or rank == "教授"): $n += 1$

D. if (gender == "男" or age < 40 or rank == "教授"): $n += 1$

D

6. 下面的程序段求 x 和 y 两个数中的大数, _____ 是不正确的。

A. $maxNum = x$ if $x > y$ else y

B. $maxNum = \text{math}/\text{max}(x, y)$

C. if ($x > y$): $maxNum = x$

D. if ($y \geq x$): $maxNum = y$

else: $maxNum = y$

$maxNum = x$ 错误

B

7. 下面的 if 语句统计“成绩(score)优秀的男生以及不及格的男生”的人数, 正确的语句为_____。

姓名:

班级:

学号:

年 月 日

C

- A. if (gender=="男" and score < 60 or score >= 90): n+=1
- B. if (gender=="男" and score < 60 and score >= 90): n+=1
- C. if (gender=="男" and (score < 60 or score >= 90)): n+=1
- D. if (gender=="男" or score < 60 or score >= 90): n+=1

and or

B

8. 用 if 语句表示如下分段函数:

$$y = \begin{cases} x^2 - 2x + 3 & x < 1 \\ \sqrt{x-1} & x \geq 1 \end{cases}$$

下面程序段不正确的是_____。

- A. if (x < 1): y = x * x - 2 * x + 3
else: y = math.sqrt(x-1)
- B. if (x < 1): y = x * x - 2 * x + 3
y = math.sqrt(x-1)
- C. y = x * x - 2 * x + 3
if (x >= 1): y = math.sqrt(x-1)
- D. if (x < 1): y = x * x - 2 * x + 3
if (x >= 1): y = math.sqrt(x-1)

A

9. 在 Python 中,以下 for 语句结构_____不能完成 1~10 的累加功能。

- A. for i in range(10,0): total += i
- B. for i in range(1,11): total += i
- C. for i in range(10,0,-1): total += i
- D. for i in (10,9,8,7,6,5,4,3,2,1): total += i

二、填空题

1. 迭代器是一个对象,表示可迭代的数据集合,包括方法 __iter__ 和 __next__,可实现迭代功能。

2. 在 Python 无穷循环“while True:”的循环体中,可以使用 break 语句退出循环。

3. Python 语句“for i in range(1,21,5): print(i, end=' ')”的输出结果为_____。

1 6 11 16

4. Python 语句“for i in range(10,1,-2): print(i, end=' ')”的输出结果为_____。

10 8 6 4 2

5. Python 循环语句 for i in range(-3,21,4) 的循环次数为 6 次。

(stop - start) // step

姓名：

班级：

学号：

年 月 日

6. 要使 Python 语句 for i in range(, -4, -2) 循环执行 15 次, 则循环变量 i 的初值应当为 26。

7. 执行下列 Python 语句后的输出结果是 /, 循环执行了 / 次。

```

i = -1;
while (i < 0): i * = i
print(i)

```

三、思考题

1. 说明以下 3 个 if 语句的区别。

<p>(1) if (i > 0):</p> <p><i>n=1 i>0, j>0</i> if (j > 0): n = 1 else: n = 2 <i>n=2 i>0, j<=0</i></p> <p>(3) if (i > 0): n = 1 else: if (j > 0): n = 2</p> <p><i>n=1 i>0</i> <i>n=2 i<=0, j>0</i></p>	<p>(2) if (i > 0): if (j > 0): n = 1 else: n = 2</p> <p><i>{ n=1 i>0, j>0</i> <i>n=2 i<=0</i></p>
---	---

2. 下列 Python 语句的运行结果为 0 1 2。

```

for i in range(3): print(i, end=' ') 0 1 2
for i in range(2, 5): print(i, end=' ') 2 3 4

```

3. 阅读下面的 Python 程序, 请问程序的功能是什么?

```

import math
n = 0
for m in range(101, 201, 2): 101~200 所有奇数
    k = int(math.sqrt(m))
    for i in range(2, k+2):
        if m % i == 0: break
    if i == k+1: 判断素数个数
        if n % 10 == 0: print()
        print('%d' % m, end=' ')
        n += 1

```


姓名：

班级：

学号：

年

月

日

6. 阅读下面的 Python 程序, 请问输出结果是什么? 程序的功能是什么?

```
print("1~1000 所有的完数有, 其因子为: ")
for n in range(1, 1001):
    total = 0; j = 0; factors = [ ]
    for i in range(1, n):
        if(n % i == 0):
            factors.append(i); total += i
    if(total == n): print("{0}: {1}".format(n, factors))
```

7. 阅读下面的 Python 程序, 请问输出结果是什么? 程序的功能是什么?

```
m = int(input("请输入整数 m: ")); n = int(input("请输入整数 n: "))
while(m != n):
    if (m > n): m = m - n
    else: n = n - m
print(m)
```

最大公约数?

11	7
4	7
4	3
1	3
1	2
1	1

第 4 章习题

一、单选题

- A 1. 下列数据类型中,Python 不支持的是_____。
A. char B. int C. float D. list
- A 2. Python 语句 `print(type(1j))` 的输出结果是_____。
A. `<class 'complex'>` 复数 B. `<class 'int'>`
C. `<class 'float'>` D. `<class 'dict'>` 字典
- C 3. Python 语句 `print(type(1/2))` 的输出结果是_____。
A. `<class 'int'>` B. `<class 'number'>`
C. `<class 'float'>` D. `<class 'double'>`
- A 4. Python 语句 `print(type(1//2))` 的输出结果是_____。
A. `<class 'int'>` B. `<class 'number'>`
C. `<class 'float'>` D. `<class 'double'>`
- B 5. Python 语句序列“`a=121+1.21; print(type(a))`”的输出结果是_____。
A. `<class 'int'>` B. `<class 'float'>`
C. `<class 'double'>` D. `<class 'long'>`
- D 6. Python 语句 `print(0xA + 0xB)` 的输出结果是_____。
A. `0xA + 0xB` B. `A + B` C. `0xA0xB` D. 21
- A 7. Python 语句序列“`x='car'; y=2; print(x+y)`”的输出结果是_____。
A. 语法错 B. 2 C. 'car2' D. 'carcar'
- D 8. Python 表达式 `sqrt(4) * sqrt(9)` 的值为_____。
A. 36.0 B. 1296.0 C. 13.0 D. 6.0
- C 9. 关于 Python 中的复数,下列说法错误的是_____。
A. 表示复数的语法是 `real+image j` ✓
B. 实部和虚部都是浮点数 ✓
C. 虚部必须后缀 `j`,且必须是小写 ~~小写~~
D. 方法 `conjugate` 返回复数的共轭复数 ✓
- A 10. Python 语句 `print(chr(65))` 的运行结果是_____。
A. 65 B. 6 C. 5 D. A
- D 11. 关于 Python 字符串,下列说法错误的是_____。
A. 字符即长度为 1 的字符串
B. 字符串以 `\0` 标志字符串的结束 ✓
C. 用户既可以用单引号,也可以用双引号创建字符串 ✓
D. 在三引号字符串中可以包含换行回车等特殊字符
- A 12. 下面 Python 程序的执行结果是_____。
`a = 10.99`
`print(complex(a))`
A. `(10.99+0j)` B. `10.99+j` C. 10.99 D. 0.99

姓名:

班级:

学号:

年

月

日

- C 13. 对于 Python 中的 random 模块,其 seed(a)函数的作用是_____。
- A. 生成一个[0.0, 1.0)之间的随机小数
 - B. 生成一个 k 比特长度的随机整数
 - C. 设置初始化随机数种子 a
 - D. 生成一个随机整数

- D 14. 运行以下 Python 程序不可能产生的输出结果是_____。
- ```
from random import *
x = [30,45,50,90]
print(choice(x))
```

- A. 30      *随机返回*      B. 45      C. 90      D. 55

- D 15. 以下关于 Python 中 random 模块的描述,正确的是\_\_\_\_\_。
- A. 设定相同种子,每次调用随机函数生成的随机数不相同 *相同*
  - B. 通过 from random import \* 导入 random 随机模块的部分函数 *全部*
  - C. uniform(0,1)与 uniform(0.0,1.0)的输出结果不同,前者输出随机整数,后者输出随机小数

- D 16. 运行以下 Python 程序不可能产生的输出结果是\_\_\_\_\_。

- ```
from random import *
print(round(random(),2))
```
- A. 0.47 B. 0.54 C. 0.27 D. 1.87
- round 四舍五入 (x, y) 操作数做 random() 生成 [0,1) 之间*

- D 17. 运行以下 Python 程序不可能产生的输出结果是_____。
- ```
from random import *
print(sample({1,2,3,4,5},2))
```

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

## 二、填空题

1. Python 内置的 4 种数值类型为 bool, float, complex, int

2. Python 内置的序列数据类型包括 list, tuple, str, bytes, bytearray

3. Python 表达式  $10 + 5 // 3 - True + False$  的值为 10。

4. Python 表达式  $3 ** 2 ** 3$  的值为 6561。

*指数: 右到左*

5. Python 表达式  $17.0 / 3 ** 2$  的值为 1.889。

*17.0 / 9*

姓名：

班级：

学号：

年 月 日

6. Python 表达式  $0 \text{ and } 1 \text{ or not } 2 < \text{True}$  的值为 0。

7. Python 语句 `print(pow(-3, 2), round(18.67, 1), round(18.67, -1))` 的输出结果是\_\_\_\_\_。

9, 18.7, 20

8. Python 语句 `print(round(123.84, 0), round(123.84, -2), floor(15.5))` 的输出结果是\_\_\_\_\_。

124, 100, 15

9. Python 语句 `print(int('20', 16), int('101', 2))` 的输出结果是\_\_\_\_\_。

32, 5

10. Python 语句 `print(hex(16), bin(10))` 的输出结果是 0x10, 1010。

11. Python 语句 `print(2.5.as_integer_ratio())` 的输出结果是 (5, 2)。

12. Python 语句 `print(float.as_integer_ratio(1.5))` 的输出结果是 (3, 2)。

13. Python 语句 `print(gcd(12, 16), divmod(7, 3))` 的输出结果是 4, (2, 1)。

14. Python 语句 `print((2-3j).conjugate() * complex(2, 3))` 的输出结果是  $(2+3j)(2+3j) = -5+12j$ 。

15. Python 语句 `print(sum((1, 2, 3)), sum((1, 2, 3), 10))` 的输出结果是\_\_\_\_\_。

6, 16

16. Python 语句 `print(abs(-3.2), abs(1-2j))` 的输出结果是\_\_\_\_\_。

3.2,  $\sqrt{5}$

17. Python 的标准随机数生成器模块是 random。



28. Python 语句 `print(chr(ord('B')))` 的执行结果是\_\_\_\_\_。

29. Python 语句 `print("hello" 'world')` 的执行结果是\_\_\_\_\_。

30. 以 123 为随机数种子,随机生成 10 个 1(含)~999(含)中的随机数,以逗号分隔打印输出,请补充横线处缺失的 Python 代码。

```
import random

for i in range(_____
print(_____, end = ",")
```

### 三、思考题

1. Python 包括哪 4 种内置的数值类型?

`int float complex bool`

2. Python 包括哪些不可变序列数据类型? 哪些可变序列数据类型?

3. Python 字符串字面量有哪 4 种定义方式?

4. Python 有哪几种类型转换方式? 各自方式是如何进行类型转换的?

强调转换

5. 阅读下面的 Python 程序, 请问输出结果是什么?

```
from decimal import *
ctx = getcontext(); ctx.prec = 2; print(Decimal('1.78'))
print(Decimal('1.78') + 0); ctx.rounding = ROUND_UP
print(Decimal('1.65') + 0); print(Decimal('1.62') + 0); print(Decimal('-1.45') + 0)
print(Decimal('-1.42') + 0); ctx.rounding = ROUND_HALF_UP
print(Decimal('1.65') + 0); print(Decimal('1.62') + 0); print(Decimal('-1.45') + 0)
ctx.rounding = ROUND_HALF_DOWN; print(Decimal('1.65') + 0); print(Decimal('-1.45') + 0)
```

6. 阅读下面的 Python 语句, 请问输出结果是什么? 程序的功能是什么?

```
import random
a = random.randint(100,999) # 随机产生一个 3 位整数
b = (a % 10) * 100 + (a // 10 % 10) * 10 + a // 100
print("原数 = ", a, ", 变换后 = ", b)
```

7. 下列 Python 语句的输出结果是\_\_\_\_\_。

```
print("数量{0}, 单价{1}".format(100, 285.6))
print(str.format("数量{0}, 单价{1:3.2f}", 100, 285.6))
print("数量 % 4d, 单价 % 3.3f" % (100, 285.6))
```

8. 下列 Python 语句的输出结果是\_\_\_\_\_。

```
print("1".rjust(20, " "))
print(format("121", "> 20"))
print(format("12321", "> 20"))
```

9. 下列 Python 语句的运行结果为\_\_\_\_\_。

```
x = False; y = True; z = False
if x or y and z: print("yes")
else: print("no")
```

10. 下列 Python 语句的运行结果为\_\_\_\_\_。

```
x = True; y = False; z = True;
if not x or y: print(1)
elif not x or not y and z: print(2)
elif not x or y or not y and x: print(3)
else: print(4)
```

姓名：

班级：

学号：

年 月 日

---

11. 阅读下面的 Python 程序, 请问输出结果是什么?

```
print(1 or 2, 0 or 2, False or True, True or False, False or 2, sep = ' ')\nprint(1 and 2, 0 and 2, False and 2, True and 2, False and True, sep = ' ')
```

12. 阅读下面的 Python 程序, 请问输出结果是什么?

```
print("T", end = ' ') if not 0 else print('F', end = ' ')\nprint("T", end = ' ') if 6 else print('F', end = ' ')\nprint("T", end = ' ') if "" else print('F', end = ' ')\nprint("T", end = ' ') if "abc" else print('F', end = ' ')\nprint("T", end = ' ') if () else print('F', end = ' ')\nprint("T", end = ' ') if (1, 2) else print('F', end = ' ')\nprint("T", end = ' ') if [] else print('F', end = ' ')\nprint("T", end = ' ') if [1, 2] else print('F', end = ' ')\nprint("T", end = ' ') if { } else print('F', end = ' ')\nprint("T", end = ' ') if {1, 2} else print('F', end = ' ')
```



姓名：

班级：

学号：

年 月 日

- D 14. Python 语句 `print(type([]))` 的运行结果是\_\_\_\_\_。  
A. `<class 'tuple'>` B. `<class 'dict'>`  
C. `<class 'set'>` ( ) D. `<class 'list'>`
- A 15. Python 语句 `print(type({}))` 的运行结果是\_\_\_\_\_。  
A. `<class 'tuple'>` ( ) B. `<class 'dict'>` {}  
C. `<class 'set'>` { } D. `<class 'list'>` []
- C 16. 以下不能创建字典的 Python 语句是\_\_\_\_\_。  
A. `dict1 = {}` B. `dict2 = {2: 6}`  
C. `dict3 = {[1,2,3]: "users"}` D. `dict4 = {(1,2,3): "users"}`
- C 17. 以下不能创建字典的 Python 语句是\_\_\_\_\_。  
A. `dict1 = {}` B. `dict2 = {1: 8}`  
C. `dict3 = dict([2,4],[3,6])` D. `dict4 = dict((2,4),[3,6])`
- B 18. Python 语句序列 `ss = list(set("banana")); ss.sort(); print(ss)` 的运行结果是\_\_\_\_\_。  
A. `['n', 'b', 'a']` B. `['a', 'b', 'n']`  
C. `['b', 'a', 'n', 'a', 'n', 'a']` D. `['a', 'a', 'a', 'b', 'n', 'n']`
- D 19. 假设将单词保存在变量 `words` 中,使用一个字典数据对象 `counts = {}` 统计单词出现的次数可以采用以下\_\_\_\_\_ Python 代码。  
A. `counts[words] = counts[words] + 1`  
B. `counts[words] = 1`  
C. `counts[words] = counts.get(words,1) + 1`  
D. `counts[words] = counts.get(words,0) + 1`
- C 20. Python 语句 `print('{: * ^10.4}'.format('Flower'))` 的运行结果是\_\_\_\_\_。(其中,“ ”表示空格)  
A. Flow B. Flower  
C. Flower D. \*\*\* Flow \*\*\*
- C 21. Python 语句 `print("{: .2f}".format(20 - 2 ** 3 + 10/3 ** 2 * 5))` 的运行结果是\_\_\_\_\_。  
A. 17.55 B. 67.56 C. 12.22 D. 17.56
- D 22. 以下 Python 语句的运行结果是\_\_\_\_\_。(其中,“ ”表示空格)  
`s1 = "猎豹"`  
`s2 = "超级奔跑健将"`  
`print("{0:^4}:{1:!  
A. 猎豹: 超级奔跑健将!!! B. 猎豹: 超级奔跑健将!!!  
C. 猎豹: ! 超级奔跑健将!! D. 猎豹: 超级奔跑健将!!!`
- A 23. 下列 Python 语句的运行结果是\_\_\_\_\_。(其中,“ ”表示空格)  
`>>> s = '百花齐放'`  
`>>> "{0:3}".format(s)`  
A. '百花' B. '百花齐'  
C. '百花齐放' D. ' 百花齐放'



9. Python 语句序列“`print(tuple([1,2,3]), list([1,2,3]))`”的运行结果是\_\_\_\_\_。
10. Python 列表解析表达式`[i for i in range(5) if i%2!=0]`和`[i**2 for i in range(3)]`的值分别为\_\_\_\_\_。
11. Python 语句“`first, * middles, last = range(6)`”执行后, `middles` 的值为\_\_\_\_\_；语句“`first, second, third, * lasts = range(6)`”执行后, `lasts` 的值为\_\_\_\_\_；语句“`* firsts, last3, last2, last1 = range(6)`”执行后, `firsts` 的值为\_\_\_\_\_；语句“`first, * middles, last = sorted([86, 85, 99, 88, 60, 95, 96])`”执行后, `sum(middles)/len(middles)`的值为\_\_\_\_\_。
12. 在 Python 中, 设有 `s=('a','b','c','d','e')`, 则 `s[2]` 值为\_\_\_\_\_；`s[2:4]` 值为\_\_\_\_\_；`s[:3]` 值为\_\_\_\_\_；`s[3:]` 值为\_\_\_\_\_；`s[1::2]` 值为\_\_\_\_\_；`s[-2]` 值为\_\_\_\_\_；`s[::-1]` 值为\_\_\_\_\_；`s[-2:-1]` 值为\_\_\_\_\_；`s[-2:]` 值为\_\_\_\_\_；`s[-99:-5]` 值为\_\_\_\_\_；`s[-99:-3]` 值为\_\_\_\_\_；`s[::]` 值为\_\_\_\_\_；`s[1:-1]` 值为\_\_\_\_\_。
13. 在 Python 中, 设有 `s=[1,2,3,4,5,6]`, 则 `max(s)` 值为\_\_\_\_\_；`min(s)` 值为\_\_\_\_\_；语句序列“`s[:1]=[]; s[:2]='a'; s[2:]='b'; s[2:3]=['x','y']; del s[:1]`”执行后, `s` 值为\_\_\_\_\_。
14. 在 Python 中, 设有 `s=['a','b']`, 则语句序列“`s.append([1,2]); s.extend('34'); s.extend([5,6]); s.insert(1,7); s.insert(10,8); s.pop(); s.remove('b'); s[3:]=[]; s.reverse()`”执行后, `s` 值为\_\_\_\_\_。
15. Python 语句 `print(len({}))` 的运行结果是\_\_\_\_\_。
16. Python 语句序列“`d={1:'x',2:'y',3:'z'}; del d[1]; del d[2]; d[1]='A'; print(len(d))`”的运行结果是\_\_\_\_\_。



```
n = _____
print("字符数为{}, 标点符号数为{}".format(n,m))
```

26. 输入一个正整数 n 作为财务数据,输出一个宽度为 20 字符,n 右对齐显示,带千位分隔符的效果,使用减号字符“-”填充。如果输入正整数超过 20 位,则按照真实长度输出。例如,输入正整数 n 为 1234,则输出如下结果“-----1,234”。请补充以下的 Python 程序。

```
n = int(input("请输入整数:"))
print(_____) # 按要求输出财务数据
```

27. 从键盘输入一个由 1 和 0 组成的二进制字符串 s(例如 1101),转换为十进制数输出显示在屏幕上(例如 13)。请补充以下的 Python 程序。

```
s = input() # 请输入一个由 1 和 0 组成的二进制数字字符串
d = 0
while _____ :
 d = _____ + (ord(s[0]) - ord('0'))
 s = _____
print("转换成十进制数是:{}".format(d))
```

### 三、思考题

1. 在 Python 中如何查询和替换一个文本字符串?

2. 在 Python 中如何实现 tuple 和 list 的转换?

3. 阅读下面 Python 语句,请问输出结果是什么?

```
s1 = "QQ"
s2 = "Wechat"
print("{: * < 10}{: = > 10}".format(s1,s2))
```

姓名：

班级：

学号：

年

月

日

---

4. 阅读下面 Python 语句,请问输出结果是什么?

```
n = int(input("请输入图形的行数: "))
for i in range(n,0, -1):
 print(" ".rjust(20 - i),end = '')
 for j in range(2 * i - 1):print(" * ",end = '')
 print("\n")
for i in range(1, n):
 print(" ".rjust(19 - i),end = '')
 for j in range(2 * i + 1):print(" * ",end = '')
 print("\n")
```

5. 阅读下面 Python 语句,请问输出结果是什么?

```
n = int(input("请输入上(或下)三角的行数: "))
for i in range(0,n):
 print(" ".rjust(19 - i),end = '')
 for j in range(2 * i + 1):print(" * ",end = '')
 print("\n")
for i in range(n-1, 0, -1):
 print(" ".rjust(20 - i),end = '')
 for j in range(2 * i - 1):print(" * ",end = '')
 print("\n")
```

6. 阅读下面 Python 语句,请问输出结果是什么?

```
daysOfWeek = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
print("DAYS: % s, MONTHS % s" % (daysOfWeek, months))
```

姓名：

班级：

学号：

年 月 日

---

7. 阅读下面 Python 语句,请问输出结果是什么?

```
names1 = ['Amy', 'Bob', 'Charlie', 'Daling']
names2 = names1; names3 = names1[:]
names2[0] = 'Alice';names3[1] = 'Ben'
total = 0
for ls in (names1, names2, names3):
 if ls[0] == 'Alice':
 total += 1
 if ls[1] == 'Ben':
 total += 2
print(total)
```

8. 阅读下面 Python 语句,请问输出结果是什么?

```
dict1 = {};dict1[1] = 1;
dict1['1'] = 3;dict1[1] += 2; total = 0
for k in dict1:
 total += dict1[k]
print(total)
```

9. 阅读下面 Python 语句,请问输出结果是什么?

```
d = {1:'a', 2:'b', 3:'c'}
del d[1];d[1] = 'x';
del d[2];print(d)
```

姓名：

班级：

学号：

年

月

日

---

10. 阅读下面 Python 语句,请问输出结果是什么?

```
item_counter = {}
def addone(item):
 if item in item_counter:
 item_counter[item] += 1
 else:
 item_counter[item] = 1
addone('Apple');addone('Pear');addone('apple')
addone('Apple');addone('kiwi');addone('apple')
print(item_counter)
```

11. 阅读下面 Python 语句,请问输出结果是什么?

```
numbers = {};numbers[(1,2,3)] = 1;
numbers[(2,1)] = 2
numbers[(1,2)] = 3
total = 0
for k in numbers:
 total += numbers[k]
print(len(numbers),' ',total,' ',numbers)
```

12. 阅读下面 Python 语句,请问输出结果是什么?

```
d1 = {'a':1, 'b':2}
d2 = d1
d1['a'] = 6
total = d1['a'] + d2['a']
print(total)
```

姓名：

班级：

学号：

年 月 日

---

13. 阅读下面 Python 语句,请问输出结果是什么?

```
d1 = {'a':1, 'b':2}
d2 = dict(d1)
d1['a'] = 6
total = d1['a'] + d2['a']
print(total)
```

14. 阅读下面 Python 语句,请问输出结果是什么?

```
fruits = ['pear', 'apple', 'kiwi', 'avocado', 'orange']
print("\n".join(fruits))
```

15. 阅读下面 Python 语句,请问输出结果是什么?

```
name = "happy birthday"
print("%s" % name[6:11])
name1 = name.replace(name[6], 'B')
print(name1)
```

16. 请使用各种方法判断字符变量 c 是否为字母字符(不区分大小写字母)。

姓名：

班级：

学号：

年

月

日

---

17. 请使用各种方法判断字符变量 c 是否为数字字符。

18. 请使用各种方法判断字符变量 c 是否为大写字母。

19. 请使用各种方法判断字符变量 c 是否为小写字母。

## 第 6 章 习题

### 一、单选题

- C 1. 以下关于 Python 文件的描述,错误的选项是\_\_\_\_\_。
- A. 二进制文件和文本文件的操作步骤都是“打开-操作-关闭” ✓
  - B. 在 open()打开文件之后,文件的内容并没有在内存中
  - C. open()只能打开一个已经存在的文件 *mode = x, 创建新的*
  - D. 在文件读写之后,需要调用 close()才能确保文件被保存在磁盘中
- C 2. 以下关于 Python 中 CSV 文件的描述,错误的选项是\_\_\_\_\_。
- A. CSV 文件可用于不同工具间进行数据交换
  - B. CSV 文件格式是一种通用的、相对简单的文件格式,应用于程序之间转移表格数据
  - C. CSV 文件通过多种编码表示字符 *Unicode*
  - D. CSV 文件的每一行是一维数据,可以使用 Python 中的列表数据类型表示
3. 以下关于 Python 文件‘+’打开模式的描述,正确的选项是 B。
- A. 追加写模式
  - B. 与 r/w/a/x 一同使用,在原功能基础上增加同时读写功能
  - C. 只读模式
  - D. 覆盖写模式
- D 4. 以下关于 Python 文件的描述,错误的选项是\_\_\_\_\_。
- A. readlines()函数读入文件内容后返回一个列表,元素划分依据是文本文件中的换行符
  - B. read()函数一次性读入文本文件的全部内容后,返回一个字符串
  - C. readline()函数读入文本文件的一行,返回一个字符串
  - D. 二进制文件和文本文件都是可以用文本编辑器编辑的文件
- D 5. 以下关于 Python 文件的描述,错误的选项是\_\_\_\_\_。
- A. open()函数的参数处理模式‘b’表示以二进制数据处理文件 ✓
  - B. open()函数的参数处理模式‘+’表示可以对文件进行读和写操作 ✓
  - C. readline()函数表示读取文件的下一行,返回一个字符串
  - D. open()函数的参数处理模式‘a’表示以追加方式打开文件, ~~删除已有内容~~
- D 6. Python 文件读取方法 read(size)的含义是\_\_\_\_\_。
- A. 从头到尾读取文件所有内容
  - B. 从文件中读取一行数据
  - C. 从文件中读取多行数据
  - D. 从文件中读取指定 size 大小的数据,如果 size 为负数或者空,则读取到文件结束

### 二、填空题

1. Python 语句 `print(1,2,3,4,5,sep='-',end='!')`的输出结果是\_\_\_\_\_。
- 1-2-3-4-5!*

姓名：                      班级：                      学号：                      年    月    日

2. Python 语句“for i in range(10): print(i, end=' ')”的输出结果是 0 1 2 3 4 5 6 7 8 9
  
3. 在 Python 程序中可以通过列表 sys.argv 访问命令行参数。argv[0] 为 Python 脚本名，argv[1] 为第一个参数名 argv[2] 为第二个参数名。
  
4. Python 程序使用 argparse 模块解析命名的命令行参数。
  
5. 如果在程序运行时需要提示用户输入密码，则可以使用模块 getpass，以保证用户输入的密码在控制台中不回显。
  
6. Python 语言使用 with 语句实现上下文管理协议。
  
7. 在 Python 语言中，使用 sys 模块中的 stdin、stdout 和 stderr，可以查看对应的标准输入、标准输出和标准错误流文件对象。
  
8. Python 可以使用函数 open() 打开文件。
  
9. Python 文件操作可以使用 close 方法关闭流，以释放资源。通常采用 with 语句，以保证系统自动关闭打开的流。
  
10. 在 Python 语言中，当打开随机文件后，可以使用实例方法 seek 进行定位。
  
11. 在 Python 语言中，fileinput 模块提供处理用于循环处理输入、输入重定向、管道或一个或多个文本文件的函数和辅助对象。
  
12. Python 语言可以使用 pickle 模块中提供的函数实现 Python 对象的序列化。

### 三、思考题

1. Python 程序通常可以使用哪几种方式实现交互功能？

输入  
输出

2. 在 Python 中使用 argparse 模块解析命名的命令行参数的主要步骤是什么？

(1) 导入模块 `import argparse`

(2) 创建 Argument Parser 对象 `parser = argparse.ArgumentParser()`

(3) 调用 parser 对象方法 `add_argument()`，增加要解析的命令行参数信息

(4) 调用 parser 对象方法 `parse_args()` 解析命令行参数，生成对应的列表

3. Python 内置的输入函数 `input()` 的语法格式是什么？其具体的参数含义各是什么？

`input([prompt])`

4. Python 内置的输出函数 `print()` 的语法格式是什么？其具体的参数含义各是什么？

`print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)`

↓                  ↓                  ↓                  ↓  
分隔符                  换行                  写入流                  是否强制刷新

5. 在 Python 中使用 `open()` 函数时，可以指定哪些打开文件的模式？默认的打开模式是什么？存在哪些打开文件的组合模式？

mode = 'r' 只读 (默认) read  
'w' 只写 write  
't' 文本格式 txt  
'b' 二进制  
'x' 创建新文件  
'a' 追加式  
'+' 可读可写, 搭配 r/w/a 组合使用

r, r+, rb+, 以此类推

6. 文本文件的读取和写入基本步骤是什么？

1. 读取：

打开、读取、关闭

2. 写入

打开、写入、关闭

7. 二进制文件的打开模式是什么？简述其读取和写入的基本步骤。

mode = 'b'

同上

打开 → { 读取  
          or  
          写入 } → 关闭

8. 如何随机访问文件？其打开模式是什么？ +

seek() 方法，

打开、定位、操作、关闭

9. Python 如何实现内存文本文件和内存二进制文件的读取操作？

10. 如何实现 Python 对象的序列化和反序列化？

pickle 模块

pickle.dump   序列化

pickle.load   反序列化

姓名：

班级：

学号：

年 月 日

11. 如何使用 os 模块提供的函数读取和写入文件？

读取：

写入：

12. 请问输入重定向和输出重定向的语法格式分别是什么？

程序 < 输入文件 | 程序 > 输出文件

13. 请问管道机制实现什么功能？管道操作的语法格式是什么？

可以指定一个程序的输出为另一程序的输入

程序 1 | 程序 2 | ... | 程序 n

## 第 7 章习题

### 一、单选题

- B 1. 如果在 Python 程序中没有导入相关的模块(例如,import random,import math)时,解释器将在运行时抛出\_\_\_\_\_错误。  
A. 语法                      B. 运行时                      C. 逻辑                      D. 不报错
- D 2. 如果在 Python 程序中包括零除运算,解释器将在运行时抛出\_\_\_\_\_错误信息。  
A. NameError                      B. FileNotFoundError  
C. SyntaxError                      D. ZeroDivisionError
- B 3. 如果在 Python 程序中试图打开不存在的文件,解释器将在运行时抛出\_\_\_\_\_错误信息。  
A. NameError                      B. FileNotFoundError  
C. SyntaxError                      D. ZeroDivisionError
- D 4. 在 Python 程序中对于表达式 123+'xyz',解释器将抛出\_\_\_\_\_错误信息。  
A. NameError                      B. FileNotFoundError  
C. SyntaxError                      D. TypeError
- B 5. 在 Python 程序中,假设列表 s=[1,2,3],如果语句中使用 s[3],则解释器将抛出\_\_\_\_\_错误信息。  
A. NameError                      B. IndexError                      C. SyntaxError                      D. TypeError
- C 6. 在 Python 程序中,假设字典 d={'1': 'male', '2': 'female'},如果语句中使用 d[3],则解释器将抛出\_\_\_\_\_错误信息。  
A. NameError                      B. IndexError                      C. KeyError                      D. TypeError
- C 7. 以下关于异常处理 try 语句块的说法,不正确的是\_\_\_\_\_。  
A. finally 语句中的代码段始终要保证被执行 ✓  
B. 一个 try 块后接一个或多个 except 块 ✓  
C. 一个 try 块后接一个或多个 finally 块 ~~多个~~ 1个  
D. try 块必须与 except 或 finally 块一起使用

### 二、填空题

1. Python 语言采用结构化的异常处理机制。在程序运行过程中,如果产生错误,则抛出异常;通过 try 语句来定义代码块,以运行可能抛出异常的代码;通过 except 语句,可以捕获特定的异常并执行相应的处理;通过 finally 语句,可以保证即使产生异常(处理失败),也可以在事后清理资源等。

2. 在某种特殊条件下,Python 代码中也可以创建一个异常对象,并通过 raise 语句,抛出给系统运行时。

姓名：

班级：

学号：

年 月 日

3. 在 Python 语言中使用关键字 `assert` 可以声明断言。

4. Python 解释器有调试模式和优化模式两种运行模式。使用选项 `-O` 运行时为优化模式, 此时内置只读变量 `__debug__` 为 `False`。

5. 在 Python 语言中自定义异常类一般继承于 `Exception` 或其子类。

### 三、思考题

1. Python 程序的错误通常可以分为哪 3 种类型? 请分别举例说明。

语法错误 `print (" ~`

运行时错误 `not import`

逻辑错误 `for 边界有错`

2. 简述 Python 中 `try...except...finally` 各语句的用法和作用。

`try`: 定义代码块

`except`: 捕获异常并执行相应处理

`finally`: 事后清理资源

3. 在 Python 语言中 `try` 语句一般有哪几种可能的形式?

3种, (1) `try...except...[else...]`

(2) `try...finally`

(3) `try...except...[else...][finally]`

4. 在 Python 语言中断言的主要功能是什么? 断言一般用于哪些情况?

调试程序

在调试模式下

5. Python 解释器有哪两种运行模式? 如何设置 Python 解释器的运行模式?

优化、调试

`-O` 启动优化模式

## 第 8 章习题

### 一、单选题

- D** 1. Python 语句 `print(type(lambda: None))` 的运行结果是\_\_\_\_\_。
- A. `<class 'NoneType'>`                      B. `<class 'tuple'>`  
 C. `<class 'type'>`                          D. `<class 'function'>`
- D** 2. Python 语句序列“`f = lambda x,y: x * y; f(12, 34)`”的运行结果是\_\_\_\_\_。
- A. 12                                              B. 22  
 C. 56                                              D. 408
- D** 3. Python 语句序列“`f1 = lambda x: x * 2; f2 = lambda x: x ** 2; print(f1(f2(2)))`”的运行结果是\_\_\_\_\_。
- A. 2                                              B. 4  
 C. 6                                              D. 8
- A** 4. 在 Python 中,若有 `def f1(p, ** p2): print(type(p2))`,则 `f1(1, a=2)` 的运行结果是\_\_\_\_\_。
- A. `<class 'int'>`                              B. `<class 'type'>`  
 C. `<class 'dict'>`                            D. `<class 'list'>`
- C** 5. 在 Python 中,若有 `def f1(a,b,c): print(a+b)`,则语句序列“`nums = (1,2,3); f1(* nums)`”的运行结果是\_\_\_\_\_。
- A. 语法错                      B. 6                      C. 3                      D. 1

### 二、填空题

1. Python 表达式 `eval("5/2+5%2+5//2")` 的结果是 5.5。
- $2.5 + 1 + 2$
2. 在 Python 中,如果要为定义在函数外的全局变量赋值,可以使用 global 语句,表明变量是在外面定义的全局变量。
3. 在 Python 中,变量按其作用域大致可以分为 全局、局部 和 类成员。
4. 在 Python 的 sys 模块中,函数 getrecursionlimit 和 setrecursionlimit 分别用于获取和设置最大递归次数。
5. 在 Python 中,使用内置函数 locals 和 globals 可以查看并输出局部变量和全局变量列表。

姓名：

班级：

学号：

年 月 日

### 三、思考题

1. Python 如何定义一个函数？

```
def ~:
 return ~
```

2. 什么是 lambda 函数？

匿名函数  
lambda 函数：表达式

3. 什么是递归函数？在递归函数使用过程中，为什么需要设置终止条件？

函数内部调用自身  
避免无限递归

4. 下列 Python 语句的输出结果是 24。

```
d = lambda p: p * 2
t = lambda p: p * 3
x = 2; x = d(x); x = t(x); x = d(x); print(x)
```

5. 下列 Python 语句的输出结果是 1 4 9。

```
i = map(lambda x: x**2, (1, 2, 3))
for t in i:
 print(t, end='')
```

6. 下列 Python 语句的输出结果是 'simple function'

```
def f1():
 "simple function"
 pass
print(f1.__doc__)
```

7. 下列 Python 语句的输出结果是 4 0。

```
counter = 1; num = 0 → 全局
def TestVariable():
 global counter
 for i in (1, 2, 3):
 counter += 1
 num = 10 → 局部
TestVariable()
print(counter, num)
```

姓名：

班级：

学号：

年

月

日

8. 下面 Python 程序的功能是什么？输出结果是什么？

```
def f(a, b):
 if b == 0: print(a)
 else: f(b, a % b)
print(f(9, 6))
```

计算最大公约数

3

9. 下列 Python 语句的输出结果是金石为开。

```
def aFunction():
 "精诚所至，金石为开"
 return 1
print(aFunction.__doc__[5:10])
```

10. 下列 Python 语句的输出结果是tuple。

```
def judge(param1, * param2):
 print(type(param2))
 print(param2)
judge(1, 2, 3, 4, 5)
```

11. 下列 Python 语句的输出结果是dict。

```
def judge(param1, ** param2):
 print(type(param2))
 print(param2)
judge(1, a = 2, b = 3, c = 4, d = 5)
```

12. 阅读下面的 Python 语句，请问如果输入 278，则输出结果是什么？程序的功能是什么？

```
aInt = input('请输入一个三位数:')
a, b, c = map(int, aInt)
print(a, b, c)
```

2 7 8

姓名：

班级：

学号：

年 月 日

13. 阅读下面的 Python 语句, 请问如果输入“3 4 90”, 则输出结果是什么? 程序的功能是什么?

```
import math
x = input('输入两边长及夹角(度):')
a, b, alpha = map(float, x.split())
c = math.sqrt(a**2 + b**2 - 2 * a * b * math.cos(alpha * math.pi/180))
print(c)
```

$$c = \sqrt{9 + 16 - 2 \times 3 \times 4 \times \cos(90 \times \frac{\pi}{180})}$$

$c = 2.5$

14. 阅读下面的 Python 语句, 请问输出结果是什么? 请分析递归实现的过程。

```
def recurse(n, s):
 if n == 0:
 print(s)
 else:
 recurse(n-1, n + s)
recurse(3, 0)
```

3, 0  
2, 3  
1, 5  
0, 6  
6

15. 阅读下面的 Python 程序, 请问绘制的图形是什么?

```
import turtle as t
def DrawCctCircle(n):
 t.penup()
 t.goto(0, -n)
 t.pendown()
 t.circle(n)
for i in range(20, 80, 20):
 DrawCctCircle(i)
```

20, 40, 60, 80  
4个同心圆

## 第 9 章习题

### 一、填空题

1. 面向对象的程序设计具有 3 个基本特征,即 封装、继承 和 多态性。
2. Python 语句序列“x= '123'; print(isinstance(x, int))”的运行结果为 False。
3. 在 Python 中,当创建对象后可以使用  .  运算符来调用其成员。
4. 在 Python 类体中, \_\_new\_\_ 是一个类方法,创建对象时调用,返回当前对象的一个实例,一般无须重载该方法。\_\_init\_\_ 方法即构造函数(构造方法),用于执行类的实例的初始化工作。在对象创建后调用,初始化当前对象的实例,无返回值。\_\_del\_\_ 方法即析构函数,用于实现销毁类的实例所需的操作,例如释放对象占用的非托管资源。
5. 在 Python 中,实例变量在类的内部通过 self 访问,在外部通过对象实例访问。

### 二、思考题

1. Python 如何复制一个对象?

copy()

2. Python 提供哪些特殊属性? 如何表示这些特殊属性? 它们各自的含义是什么?

--dict--

--class--

--base--

--init--

--doc--

姓名：

班级：

学号：

年 月 日

3. 下列 Python 语句的运行结果为 100 / 100。

```
class parent:
 def __init__(self, param):
 self.v1 = param
class child(parent):
 def __init__(self, param):
 super().__init__(param)
 self.v2 = param
obj = child(100); print ("%d %d" % (obj.v1, obj.v2))
```

4. 下列 Python 语句的运行结果为 100。

```
class Account:
 def __init__(self, id):
 self.id = id;
 id = 888
acc = Account(100); print(acc.id)
```

5. 下列 Python 语句的运行结果为 400。

```
class account:
 def __init__(self, id, balance):
 self.id = id;
 self.balance = balance
 def deposit(self, amount):
 self.balance += amount
 def withdraw(self, amount):
 self.balance -= amount
acc1 = account('1234', 100)
acc1.deposit(500) 100+500=600
acc1.withdraw(200) 600-200=400
print(acc1.balance)
```

6. 下列 Python 语句的运行结果为 12。

```
class A:
 def __init__(self, a, b, c):
 self.x = a + b + c
a = A(6,2,3)
b = getattr(a, 'x') x=11
setattr(a, 'x', b+1) b+1=12
print(a.x) a.x=12
```

7. 阅读下面的 Python 语句, 请问输出结果是什么? 12

```
d1 = {'a':[1,2], 'b':2}
d2 = d1.copy() → 浅拷贝, d1变, d2不变
d1['a'][0] = 6 [6,2]
total = d1['a'][0] + d2['a'][0]
print(total) 6+6=12
```

姓名：

班级：

学号：

年

月

日

8. 阅读下面的 Python 语句, 请问输出结果是什么?

```
from copy import *
d1 = {'a':[1,2], 'b':2}
d2 = deepcopy(d1)
d1['a'][0] = 6
total = d1['a'][0] + d2['a'][0]
print(total)
```

7

9. 下列 Python 语句的运行结果为 30。

```
list1 = [1,2,3]; list2 = [3,4,5]
dict1 = {'1':list1, '2':list2}
dict2 = dict1.copy()
dict1['1'][0] = 15
print(dict1['1'][0] + dict2['1'][0])
```

10. 下列 Python 语句的运行结果为 16。

```
import copy
list1 = [1,2,3]; list2 = [3,4,5]
dict1 = {'1':list1, '2':list2}
dict2 = copy.deepcopy(dict1)
dict1['1'][0] = 15
print(dict1['1'][0] + dict2['1'][0])
```

11. 下列 Python 语句的运行结果为 21。

```
class Person:
 def __init__(self, id):
 self.id = id
mary = Person(123)
mary.__dict__['age'] = 18
mary.__dict__['gender'] = 'female'
print(mary.age + len(mary.__dict__))
```

18 + 3

字典  
有 id,  
age  
gender  
共  
3

## 第 10 章习题

### 一、填空题

1. 在 Python 中包含了数量众多的模块,通过 import 语句可以导入模块,并使用其定义的功能。

2. 在 Python 中假设有模块 m,如果希望同时导入 m 中的所有成员,则可以采用 import \* 的导入形式。

3. 在 Python 中使用内置函数 importlib 也可以导入模块。

4. 在 Python 中 sys 模块的 sys.path 属性返回一个路径列表。

5. 在 Python 中每个模块都有一个名称,通过特殊变量 \_\_name\_\_ 可以获取模块的名称。特别地,当一个模块被用户单独运行时,模块名称为 '\_\_main\_\_'

6. 在 Python 模块中定义的所有成员,包括变量、函数和类,可以通过内置的函数 dir() 查询,也可以通过 help() 函数查询其帮助信息。

### 二、思考题

1. 什么是模块? 模块是如何导入 Python 解释器的? 分别有哪几种方法?

模块对应于 Python 源代码文件

import 语句

2. Python 包和模块是什么关系? 包和模块组成的层次组织结构分别对应于什么?

各个功能相似的模块可以组织成一个包,

姓名：

班级：

学号：

年

月

日

3. 在 Python 中创建包的基本步骤和内容是什么？如何导入和使用包？

- 1) 在指定目录中创建对应包名的目录
- 2) 在该目录下创建 `--init--.py` 文件
- 3) 在该目录下创建模块文件

`import ~`  
`from ~ import ~ , import ~ as ~`

4. 在 Python 中导入模块时一般采用什么搜索顺序？

- 1) 当前目录
- 2) Python 标准库
- 3) Python 扩展库

5. 在 Python 中名称空间与名称查找顺序是什么？

- 1) 局部 名称空间
- 2) 全局 ~
- 3) 内置 ~

## 第 11 章习题

### 一、填空题

1. 对长度为  $n$  的线性表进行顺序查找, 在最坏的情况下所需要的比较次数是  $n$  次。

2. 在 Python 中使用函数  $\text{sys.getsizeof}(x)$  可以返回一个内置数据类型  $x$  在系统中所占用的字节数。

3. 数据结构通常由三个部分组成, 即数据的 逻辑 结构, 数据的 存储 结构和数据的 运算 结构。

4. 对于如下的 Python 程序代码, 其运行时间为  $O(n^2)$ , 算法的时间复杂度 (增长量级) 为  $O(n^2)$ 。

```
for i in range(1, n):
 s = ""
 for j in range(1, n):
 s += str.format("{0:1} * {1:1} = {2:<2} ", i, j, i * j)
 print(s)
```

5. 在深度为 7 的满二叉树中, 节点个数总共是  $2^7 - 1$  个, 叶子节点总个数是  $2^6$  个。

### 二、思考题

1. 下列 Python 语句的执行结果是 \_\_\_\_\_。

```
from collections import *
m1 = {1:'a', 2:'b'}; m2 = {2:'a', 3:'x', 4:'y'}; m = ChainMap(m1, m2)
print(m.maps, m.parents, m.new_child())
print(m[1], m[3]); m[1] = 'A'; m[3] = 'X'; print(m)
```

2. 下列 Python 语句的执行结果是 \_\_\_\_\_。

```
from collections import *
c1 = Counter(); print(c1); c2 = Counter('banana'); print(c2)
c3 = Counter({'R': 4, 'B': 2}); print(c3)
c4 = Counter(birds = 2, cats = 4, dogs = 8); print(c4)
print(c4['flowers'], c4['cats']); print(list(c3.elements()))
print(c4.most_common(2)); c3.subtract('RGB'); print(c3)
```

3. 下列 Python 语句的执行结果是  $9$ 。

```
from collections import *
```

$2$   
 $C$   
 $C$   
 $2$   
 $a$

```
dq = deque(); dq.append('a'); dq.append(2); dq.append('c'); data = iter(dq)
while True:
```

```
 try: i = next(data)
 except StopIteration: break
 print(i, end = '')
```

```
print(dq.pop(), dq.pop(), dq.pop())
```

4. 下列 Python 语句的执行结果是 2。

```
from collections import *
```

```
dq = deque(); dq.append('a'); dq.append(2); dq.append('c')
print(dq.popleft(), dq.popleft(), dq.popleft())
```

5. 下列 Python 语句的执行结果是 (('r', 3), ('g', 2), ('b', 1))

```
from collections import defaultdict
s = [('r', 3), ('g', 2), ('b', 1)]; dd = defaultdict(int, s); print(dd['b'], dd['w'])
s1 = [('r', 3), ('g', 2), ('b', 1), ('r', 5), ('b', 4)]; dd1 = defaultdict(list)
for k, v in s1: dd1[k].append(v)
print(list(dd1.items()))
```

6. 下列 Python 语句的执行结果是 \_\_\_\_\_。

```
from collections import *
d = {'red':3, 'green':4, 'blue':1}
print(d.items(), sorted(d.items()))
od = OrderedDict(sorted(d.items()))
print(od.popitem(), od.popitem(False))
```

7. 下列 Python 语句的执行结果是 \_\_\_\_\_。

```
from collections import *
p = namedtuple('Point', ['x', 'y'])
p.x = 1; p.y = 2; print(p._fields, p.x, p.y)
t = [10, 20]; p1 = p._make(t); print(p1._asdict())
print(p1._replace(x = 100), p1.x, p1.y)
```

8. 下列 Python 语句的执行结果是 \_\_\_\_\_。

```
import array; arr1 = array.array('i', (1, 2, 3, 4, 5))
arr1[1] = 22; print(arr1, arr1[2:], type(arr1[1]))
del arr1[2:]; print(arr1, arr1.typecode, arr1.itemsize)
```

9. 下列 Python 语句的执行结果是 1 3 3 5 65 49 8, 9

```
import array
a = array.array('b', (3, 2)); a.append(3); a.extend((3, 5))
print(a, a.count(3)); a.frombytes(b'11'); a.fromlist([8, 9])
print(a, a.index(3)); a.insert(0, 1); a.pop()
a.remove(2); a.reverse(); print(a.tolist())
```

10. 一棵二叉树共有 25 个节点，其中 5 个是叶子节点，则度为 1 的节点数有多少个？

$$N_0 = N_2 + 1$$

↓  
叶      度=2的

$$\therefore n = N_0 + N_1 + N_2 = 25$$

5      4

$$\therefore 16$$

姓名：

班级：

学号：

年 月 日

## 第 12 章习题

### 一、填空题

1. Python 的标准 GUI 库 tkinter 由若干的模块组成, 例如 tkinter、tkinter 和 tkinter.constants。

2. Python 图形用户界面程序一般包含一个顶层窗口, 也称 根窗口 或 顶层窗口。

3. tkinter 提供了三种不同的几何布局管理类, 即 pack、grid 和 place, 用于组织和管理在父组件中子配件的布局方式。

4. 在 tkinter 中通过组件的 width 和 height 选项可以设置组件的宽度和高度。

5. 在 tkinter 中通过组件的 font 选项可以设置其显示的文本的字体。

6. 在 tkinter 中通过组件的 anchor 选项可以设置内容停靠位置。

7. 在 tkinter 中通过组件的 Cursor 选项可以设置鼠标经过组件时的光标形状。

8. 在 tkinter 中通过组件的 text 选项可以设置其显示的内容。通过 wrapping 选项可指定多少单位后开始换行, 即显示多行; 通过 justify 选项可指定多行的对齐方式。

9. 在 tkinter 中通过组件的 bitmap 选项可以设置其显示的位图。自定义位图为 xbm 格式的文件。

10. 在 tkinter 中通过组件的 image 选项可以设置其显示的图像。

11. 在 tkinter 中通过组件的 compound 选项可以设置其同时显示文本和位图/图像。

姓名：                      班级：                      学号：                      年    月    日

12. 在 tkinter 中通过组件的 relief 选项可以设置其 3D 显示样式。通过 active relief 选项可以设置其鼠标经过时的 3D 显示样式。

13. 在 tkinter 中通过组件的 boderwidth 或 bd 选项可以设置其边框宽度。

14. 在 tkinter 中通过组件的 padx 和 pady 选项可以设置其显示内容与边框之间的填充宽度和高度。

15. 在 tkinter 中通过组件的 State 选项可以设置其启用或禁用状态。

16. 在 tkinter 中通过组件的 underline 选项可以设置组件显示文本第几个字符加下画线。

17. tkinter 通过组件的 textvariable 选项可以绑定 StringVar 对象到组件。

18. 在 tkinter 中 Radiobutton 组件用于选择同一组单选按钮中的一个单选按钮(不能同时选定多个),可显示文本,也可显示图像。

19. 在 tkinter 中 Checkbutton 组件用于选择一项或多项选项(可以同时选定多个),可显示文本,也可显示图像。

20. 在 tkinter 中 Listbox 组件用于显示对象列表,并且允许用户选择一个或多个项。

21. 在 tkinter 中 Combobox 组件允许用户选择一个项的列表框(在用户请求时显示)。用户单击下拉按钮可显示列表框,选择的内容会显示在顶部文本框中。

22. 在 tkinter 中 Scale 组件用于在有界区间内,通过移动滑块来选择值。

23. 位于 tkinter 模块中的子模块 `messagebox`、`filedialog`、`colorchooser` 和 `simpledialog` 包括通用的预定义对话框；用户也可以通过继承 `TopLevel` 创建自定义对话框。

24. 位于 tkinter 模块中的子模块 `messagebox` 用于实现通用消息对话框的功能。

25. 位于 tkinter 模块中的子模块 `filedialog` 用于实现文件对话框的功能。

26. 位于 tkinter 模块中的子模块 `colorchooser` 用于实现颜色选择对话框的功能。

27. 位于 tkinter 模块中的子模块 `simpledialog` 用于实现输入对话框的功能。

## 二、思考题

1. 在 Python 中有哪几种导入 tkinter 模块的方法？

```
import tkinter as tk
import tkinter
from tkinter import *
```

2. 在 Python 中包括哪些常用的组件？

基础组件：Label（标签）、Button（按钮）、Entry（输入框）、Text（文本框）。

选择组件：Radiobutton（单选）、Checkbutton（多选）、Listbox（列表框）、Combobox（下拉列表）。

容器组件：Frame（框架）、Canvas（画布）。

交互组件：Scale（滑块）、Scrollbar（滚动条）。

3. Python 图形用户界面应用程序包括哪几种类型的菜单？

主菜单  
上下文菜单  
下拉菜单

姓名：

班级：

学号：

年

月

日

4. 在 Python 中创建主菜单一般遵循哪些步骤？

1. 创建主菜单对象 `menu = tk.Menu(root)`

2. 向主菜单添加子菜单

3. 向子菜单添加菜单项

4. 将主菜单绑定到根窗口

5. 在 Python 中创建上下文菜单一般遵循哪些步骤？

1. `context-menu = tk.Menu(root, tearoff=0)`

2. 添加

3. 绑定事件

## 第 13 章习题

### 一、单选题

1. 在 matplotlib.pyplot 模块中绘制折线图,可使用以下 \_\_\_\_\_ 绘制函数。  
A.  plot()      B. bar()      C. pie()      D. scatter()
2. 在 matplotlib.pyplot 模块中绘制散点图,可使用以下 \_\_\_\_\_ 绘制函数。  
A. plot()      B. bar()      C. pie()      D.  scatter()
3. 在 matplotlib.pyplot 模块中绘制条形图,可使用以下 \_\_\_\_\_ 绘制函数。  
A. plot()      B.  bar()      C. pie()      D. scatter()
4. 在 matplotlib.pyplot 模块中绘制柱形图,可使用以下 \_\_\_\_\_ 绘制函数。  
A. plot()      B.  bar()      C. pie()      D. scatter()
5. 在 matplotlib.pyplot 模块中绘制饼图,可使用以下 \_\_\_\_\_ 绘制函数。  
A. plot()      B. bar()      C.  pie()      D. scatter()
6. 在 matplotlib.pyplot 模块中绘制直方图,可使用以下 \_\_\_\_\_ 绘制函数。  
A.  hist()      B. bar()      C. pie()      D. scatter()

### 二、填空题

1. Python 标准库中包括图形绘制相关模块,其中,模块 tkinter 用于画布绘图,模块 turtle 用于海龟绘图。

2. Matplotlib 是 Python 最著名的绘图库之一,其 pyplot 子库(子模块)提供了和 MATLAB 类似的绘图 API,方便用户快速绘制 2D 图表,包括直方图、饼图、散点图等。

3. 在 matplotlib.pyplot 模块中使用函数 plt.show() 可显示绘制的图形。

4. 在 matplotlib.pyplot 模块中使用函数 plt.savefig() 可保存绘制的图形。

5. 在 matplotlib.pyplot 模块中使用函数 plt.title() 可设置图表的标题。

6. 在 matplotlib.pyplot 模块中使用函数 plt.ylim() 可设置 y 轴坐标的范围。

7. 在 matplotlib.pyplot 模块中使用函数 plt.ylabel() 可设置 y 轴坐标的标题。