

青少年软件编程 (Python) 等级考试试卷 (六级)

分数：100 题数：38

一、单选题(共 25 题，共 50 分)

1. 在 Python 中，打开文件并读取所有内容的正确方法是？ ()

```
with open('file.txt', __) as f:  
    content = f.read()
```

- A. 'r'
- B. 'w'
- C. 'a'
- D. 'x'

2. 在 Python 中，以下代码的执行结果是？ ()

```
with open('test.txt', 'w+') as f:  
    f.write('HelloWorld')  
    f.seek(5)  
    print(f.read(2))
```

- A. He
- B. Wo
- C. or
- D. 报错

3. 在 Python 中，以下关于文件追加操作的描述，正确的是？ ()

- A. 'a+'模式允许读取和追加，指针初始位置在文件开头
- B. 'a'模式下写入内容会覆盖文件开头的内容
- C. 'w'模式和'a'模式在文件不存在时都会创建新文件
- D. 追加操作不能与 seek()配合使用

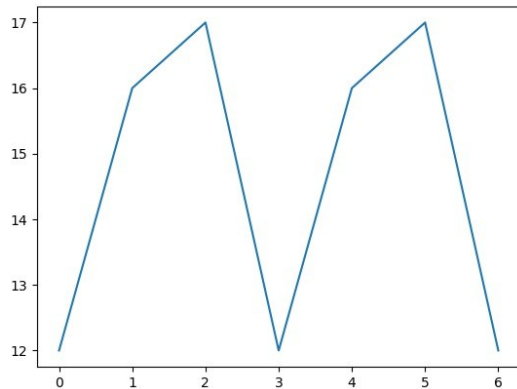
4. 在 Python 中，以下哪个选项可以正确写入内容到文件？ ()

```
with open('file.txt', 'w') as f:  
    f.___('Hello World')
```

- A. save
- B. put
- C. write
- D. add

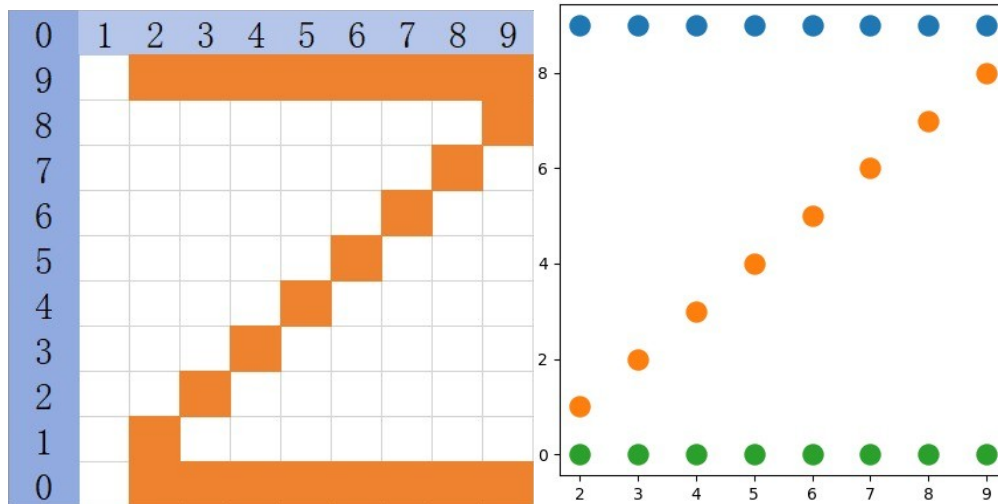
5. 编写程序绘制如下图形，①处可以填入的内容是？ ()

```
import numpy as np  
from matplotlib import pyplot as plt  
x = np.arange( __①__ )  
y = [12,16,17,12,16,17,12]  
plt.plot(x, y)  
plt.show()
```



- A. 1,7
- B. 8
- C. 7
- D. 9

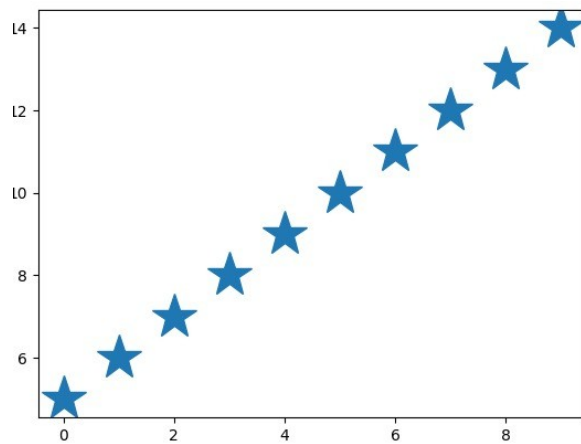
6. 已知字母 Z 的点阵和坐标如下图所示，编写程序用绘制点的方式描出字母 Z，关于程序说法正确的是？（ ）



```
import matplotlib.pyplot as plt
x1=[2,3,4,5,6,7,8,9]
y1=[9,9,9,9,9,9,9,9]
x2=①
y2=[1,2,3,4,5,6,7,8]
x3=[2,3,4,5,6,7,8,9]
y3=[0,0,0,0,0,0,0,0]
plt.scatter(x1,y1,marker='o',s=150)
plt.scatter(x2,y2,marker='o',s=150)
plt.scatter(x3,y3,marker='o',s=150)
plt.show()
```

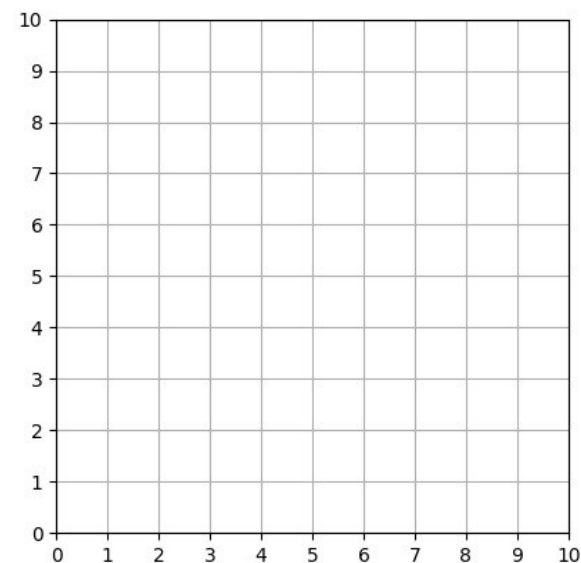
- A. ① 代码为：[2,3,4,5,6,7,8,9]
- B. ① 代码为：[1,2,3,4,5,6,7,8]

- C. x_1, x_2, x_3 的所有值都减少 1 时，整个字母将变宽 1
- D. 从 y_1, y_2, y_3 的值可以看出，当它们相等时，将绘制出垂直线
7. 绘制一条用星星组成的直线，程序说法错误的是？ ()



```
import numpy as np
from matplotlib import pyplot as plt
x = np.arange(10)
y = x + 5
plt.plot(x, y, '①', markersize=30)
plt.show()
```

- A. ① 处内容是*
- B. 语句 `x = np.arange(10)` 中的 10 表示星星的数量
- C. 语句 `plt.plot(x, y, '*', markersize=30)` 中的 `markersize` 表示星星的大小
- D. 语句 `x = np.arange(10)` 可以用 `x = int( range(10))` 替换
8. 关于绘制如下坐标图的程序，说法错误的是？ ()



```
import numpy as np
import matplotlib.pyplot as plt
```

```
plt.xlim(0,①)
plt.xticks(list(range(0,11)))
plt.ylim(0,10)
plt.yticks(list(range(0,11)))
ax = plt.gca()
ax.set_aspect('equal')
ax.grid(True)
plt.show()
```

- A. 语句 ax.grid(True)与语句 ax.grid(1)是等效的
- B. 语句 plt.yticks(list(range(0,11)))与语句 plt.yticks(tuple(range(0,11)))是等效的
- C. ① 处内容为 10
- D. 运行程序后，当用鼠标调节生成图片的窗口宽度时，图片会变形（宽和高比例发生变化）

9. 在 Python 中，下面的选项中能查询 users 表前 10 条记录的 SQL 是？（ ）

- A. SELECT 10 * FROM users
- B. SELECT TOP 10 FROM users
- C. SELECT * FROM users LIMIT 10
- D. SELECT * FROM users WHERE ROWNUM <= 10

10. 在 Python 中，下面代码可以正确连接 SQLite 数据库的是？（ ）

- A. conn = sqlite3.open('test.db')
- B. conn = sqlite3.connect('test.db')
- C. conn = sqlite3.create('test.db')
- D. conn = sqlite.connect('test.db')

11. 在 Python 的 SQLite 中，用于执行 SQL 语句的方法是？（ ）

- A. execute()
- B. fetch()
- C. run()
- D. query()

12. 由下列 Python 代码：

```
import sqlite3
conn = sqlite3.connect("d:/stu.db")
cur = conn.cursor( )
cur.execute("SELECT * FROM students")
data = cur.fetchall()
cur.close()
conn.close()
```

可知当前的数据表名是？（ ）

- A. cur
- B. stu.db
- C. students
- D. data

13. 如果要将以下 Python 数据结构保存成 JSON 格式，应该使用哪个方法？（ ）

```
student = {"name": "张三", "age": 15, "score": 95}
```

- A. json.dump()
- B. json.read()
- C. json.write()
- D. json.open()

14. 在处理 CSV 文件时，下面哪种文件打开模式最常用？（ ）

- A. 'r' 或 'w'，并加上 newline="" 参数
- B. 'a'，并加上 encoding='utf-8'
- C. 'b'，二进制模式
- D. 'r+'，可读写模式

15. 下列关于 JSON 格式的描述，正确的是？（ ）

- A. JSON 文件只能存储字符串
- B. JSON 文件不能嵌套列表
- C. JSON 数据和 Python 字典结构相似
- D. JSON 只能用 Python 读写

16. 假如有以下二维列表：

```
data = [  
    ["姓名", "数学", "英语"],  
    ["张三", 90, 85],  
    ["李四", 80, 70]  
]
```

使用 Python 写入 data.csv 文件，应该选择的代码是？（ ）

- A. json.dump(data, open('data.csv', 'w'))
- B. csv.writer(open('data.csv', 'w')).writerows(data)
- C. open('data.csv', 'w').write(data)
- D. data.to_csv('data.csv')

17. 假设你读取了一个 CSV 文件，内容如下：

姓名,成绩

小明,90

小红,85

想将它读成字典，应该使用 csv 库中的哪个方法最合适？（ ）

- A. csv.reader()
- B. csv.writer()
- C. csv.DictReader()

D. csv.load()

18. 下列有关类和对象的概念，说法正确的是？（ ）

- A. 通过 def 关键字定义类
- B. 通过 class 创建实例
- C. 每个实例的数据相同
- D. 每个实例拥有相同的方法

19. 你在开发一个宠物管理系统，使用面向对象的方式来表示不同的动物，运行后输出的结果是？（ ）

```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        return "发出声音"
class Dog(Animal):
    def speak(self):
        return f"{self.name} 汪汪叫"
dog = Dog("旺财")
print(dog.speak())
```

- A. 发出声音
- B. 旺财 发出声音
- C. 旺财 汪汪叫
- D. 代码报错

20. 在一个学校管理系统中，有一个基类 Person 表示人员，有属性 name 和 age，还有一个子类 Student 继承自 Person，并额外有属性 student_id。以下代码的空白处应填入什么来正确初始化子类实例？（ ）

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
class Student(Person):
    def __init__(self, name, age, student_id):
        _____
        self.student_id = student_id
# 创建学生实例
student = Student("小明", 18, "001")
```

- A. super().__init__(name, age)
- B. Person.__init__(name, age)
- C. self.__init__(name, age)
- D. super().init(name, age)

21. 有如下 Python 程序代码：

```
class Person(): # ①
    def __init__(self, name, age): # ②
        self.name = name # ③
        self.age = age
zs = Person("张三", "15") # ④
```

下列说法正确的是？（ ）

- A. ①处 Person 的“P”可以改为小写
- B. ②处的 self 不可以省略
- C. ③处 self.name 为实例
- D. ④处是调用函数

22. 在 tkinter 中，以下哪个方法用于设置窗口的大小？（ ）

- A. win.size()
- B. win.title()
- C. win.geometry()
- D. win.resize()

23. 在 tkinter 中，按钮（Button）添加到窗口后，若要让它显示，必须调用哪个方法？（ ）

- A. show()
- B. pack()
- C. display()
- D. enable()

24. 在设计用户登录界面时，输入框用哪个组件实现用户信息输入？（ ）

- A. Label
- B. Button
- C. Entry
- D. TextArea

25. 下列关于布局管理器的描述，正确的是？（ ）

- A. pack()、place()、grid()都能控制组件位置
- B. 只有 pack()能用来放置组件
- C. grid()只能放在按钮上使用
- D. place()自动均匀排列组件

二、判断题(共 10 题，共 20 分)

26. 在 Python 中，下面这段代码可以正确读取二进制文件并获取前 10 个字节。（ ）

```
with open('data.bin', 'r', encoding='utf-8') as f:
    data = f.read(10)
```

27. 在 Python 语言中，语句 numpy.arange(10)和 range(10)生成的序列是相类似的。（ ）

28. 在 Python 的 matplotlib 库中，plt.bar()函数可用来绘制散点图。（ ）

29. 在 Python 语言的 SQLite 中，DROP 语句用于删除表中的记录。（ ）
30. JSON 格式的数据可以直接通过 Python 的字典操作进行修改。（ ）
31. 二维列表可以直接通过 open().write()方法写入文本文件，不需要额外处理。（ ）
32. 在 Python 里，当子类继承父类后，子类只能使用自己定义的属性和方法，不能访问父类的属性和方法。（ ）
33. 某车辆租赁系统中有如下代码段；
- ```
class Vehicle:
 def start_engine(self):
 return "引擎启动"
class ElectricCar(Vehicle):
 def start_engine(self):
 return "电机启动"
car = ElectricCar()
print(car.start_engine()) # 输出: 电机启动
```
- 在车辆租赁系统中，ElectricCar 类通过重写 Vehicle 类的 start\_engine 方法，实现了子类对父类方法的覆盖，这体现了面向对象中“继承允许子类定制父类行为”的特性。（ ）
34. 在 Python 语言的 tkinter 中，Label 组件只能用来显示文字，不能用来显示图片。（ ）
35. Python 中使用'w'模式打开文件时，如果文件已存在会抛出异常。（ ）
- 三、编程题(共 3 题，共 30 分)
36. 数据合并

如下图所示，学校食堂要将大米的入库和出库数据，合并到文件“合并.csv”中。其中入库数据文件为“入库.csv”，出库数据文件为“出库.csv”，文件“入库.csv”、文件“出库.csv”和“合并.csv”都放置在程序所在的文件夹，请补全代码。

|    |           |      |    |           |      |    |           |      |
|----|-----------|------|----|-----------|------|----|-----------|------|
|    |           |      |    |           |      | 1  | 入库时间      | 入库数量 |
|    |           |      |    |           |      | 2  | 2023/2/12 | 2    |
|    |           |      |    |           |      | 3  | 2023/2/17 | 8    |
|    |           |      |    |           |      | 4  | 2023/2/23 | 10   |
|    |           |      |    |           |      | 5  | 2023/4/21 | 5    |
|    |           |      |    |           |      | 6  | 2023/5/12 | 10   |
|    |           |      |    |           |      | 7  | 2023/5/19 | 10   |
|    |           |      |    |           |      | 8  | 2023/5/26 | 10   |
|    |           |      |    |           |      | 9  | 2023/9/1  | 15   |
|    |           |      |    |           |      | 10 | 2023/9/8  | 10   |
|    |           |      |    |           |      | 11 | 2023/9/15 | 12   |
| 1  | 入库时间      | 入库数量 | 1  | 出库时间      | 出库数量 | 12 | 出库时间      | 出库数量 |
| 2  | 2023/2/12 | 2    | 2  | 2023/1/2  | 2    | 13 | 2023/1/2  | 2    |
| 3  | 2023/2/17 | 8    | 3  | 2023/1/3  | 2    | 14 | 2023/1/3  | 2    |
| 4  | 2023/2/23 | 10   | 4  | 2023/1/4  | 2    | 15 | 2023/1/4  | 2    |
| 5  | 2023/4/21 | 5    | 5  | 2023/1/5  | 2    | 16 | 2023/1/5  | 2    |
| 6  | 2023/5/12 | 10   | 6  | 2023/2/13 | 1    | 17 | 2023/2/13 | 1    |
| 7  | 2023/5/19 | 10   | 7  | 2023/2/14 | 1    | 18 | 2023/2/14 | 1    |
| 8  | 2023/5/26 | 10   | 8  | 2023/2/20 | 2    | 19 | 2023/2/20 | 2    |
| 9  | 2023/9/1  | 15   | 9  | 2023/2/21 | 2    | 20 | 2023/2/21 | 2    |
| 10 | 2023/9/8  | 10   | 10 | 2023/2/22 | 2    | 21 | 2023/2/22 | 2    |
| 11 | 2023/9/15 | 12   | 11 | 2023/2/23 | 2    | 22 | 2023/2/23 | 2    |

```

import csv
in1=[]
in2=[]
with open('/data/入库.csv') as f1:
 rows = list(____①____)
 for a in rows:
 in1. ____②____
with open('/data/出库.csv') as f2:
 rows = list(csv.reader (f2))
 for a in rows:
 in2. ____③____
with open('合并.csv', 'w', encoding='utf_8_sig', newline='') as f3:
 writer = csv.writer(f3)
 for a in in1:
 writer.writerow(a)
 for a in in2:
 writer. ____④____

```

### 37. 存取款操作

创建一个名为"BankAccount"的类，要求：

- (1) 有构造函数(init)，接受账户名和初始余额；

- (2) 有 deposit(amount)方法用于存款；
  - (3) 有 withdraw(amount)方法用于取款（余额不足时提示）；
  - (4) 有 display()方法显示账户信息；
  - (5) 创建一个实例，进行存款、取款操作，并显示最终账户信息。
- 请根据要求，补全代码。

```
class BankAccount:
 def ____①____ (self, account_name, initial_balance):
 self.account_name = account_name
 self.balance = ____②____

 def deposit(self, amount):
 if amount > 0:
 ____③____
 print(f"成功存入 {amount} 元")
 else:
 print("存款金额必须大于 0")

 def withdraw(self, amount):
 if amount > self.balance:
 print("余额不足")
 else:
 ____④____
 print(f"成功取出 {amount} 元")

 def display(self):
 print(f"账户名：{self.account_name}")
 print(f"当前余额：{ ____⑤____ }元")

account = BankAccount("张三", 1000)
account.deposit(500)
account.withdraw(200)
account.withdraw(2000) # 测试余额不足情况
account.display()
```

### 38. 成绩数据库

现有学生成绩数据库 school.db，包含表结构：

```
CREATE TABLE scores (id INT, name TEXT, math INT, class TEXT)
```

请补全代码完成以下功能：

1. 修改指定班级学生的数学成绩
2. 统计各班数学成绩 > 120 分的学生人数
3. 输出格式化结果

(本题无需运行通过，写入代码即可)

```
import sqlite3
```

```
conn = sqlite3.connect('school.db')
cursor = conn.cursor()
```

# 任务 1：修改成绩（示例：将 1 班张三数学成绩改为 135）

```
update_sql = "UPDATE scores SET math = ? WHERE class = ? AND name = ?"
```

```
cursor.execute(update_sql, (135, '1 班', '张三'))
```

①

# 任务 2：统计各班数学 120 高分人数

```
cursor.execute("SELECT class FROM scores WHERE math > 120")
```

```
results = _____ ②
```

```
class_counts = {}
```

```
for row in results:
```

```
 class_name = row[0]
```

```
 if class_name in class_counts:
```

```
 _____ ③
```

```
 else:
```

```
 class_counts[class_name] = 1
```

# 任务 3：格式化输出结果

```
print("各班数学高分统计：")
```

```
for class_name, count in _____ ④ :
```

```
 print(f"{class_name.ljust(6)}: {'★'*count}")
```

```
cursor.close()
```

```
_____ ⑤
```