

2026 年信息素养大赛 Python 模拟卷二答案

- 1.B. float
- 2.B. 总载重量: 3750 斤
- 3.A. if caravan_size > 50: print(“需特别通报”)
- 4.C. stations
- 5.A. 需要 15 个箱子, 最后一个箱子装 8 件
- 6.C. 函数必须有一个 return 语句。
- 7.B. 使用一个字典 (dict), 键为城市名, 值为数量。
- 8.C. caravan_size[2]
- 9.B. 1415
- 10.C. total += num
- 11.B. import math
- 12.D. camel_count
- 13.D. bool
- 14.A. 香料: 30 袋, 丝绸: 0 匹
- 15.B. total_tax = (price + freight) * tax_rate
16. (×)
17. (×)
18. (×)
19. (√)
20. (√)

详细解析:

1. 本题正确答案为 B. float。

详细解析:

题目要求记录可能为小数的重量 (如 12.5 斤), 因此需要能存储小数的数据类型。

A 选项 int 只能存整数, 会丢失精度;

B 选项 float 可存小数和整数, 适合数值计算;

C 选项 str 是字符串, 不能直接运算;

D 选项 bool 只有两个值, 无法表示重量。

结论: 应使用 float。

2. 本题正确答案为 B. 总载重量: 3750 斤。

详细解析:

代码计算 15 头骆驼 × 250 斤/头 = 3750 斤, 存入 total_load。

print 语句用 format 将 3750 填入字符串, 输出 “总载重量: 3750 斤”。

错误选项 A 误做加法(265), C 只取单驼载重(250), D 只取数量(15)。

结论: 正确结果为 B。

3. 本题正确答案为 A. if caravan_size > 50 :print (“需特别通报”)

详细解析：

题目要求判断人数是否超过 50，需使用 Python 的条件语句。

A 选项语法正确，条件后跟冒号，然后执行打印；

B 选项多了 `then`，Python 中不允许；

C 选项 `when` 不是 Python 关键字；

D 选项的大括号和括号是 C/Java 的写法，不符合 Python 语法。

结论：正确选项为 A。

4. 本题正确答案为 C. `stations`

详细解析：

题目要求遍历驿站列表并输出每个驿站名称。`for` 循环需要填入可迭代对象，

C 选项直接使用列表 `stations`，每次循环取出一个驿站名赋值给 `station`。

A 选项 5 是整数，不能迭代；

B 选项 `range(5)` 生成 0~4 的数字，不是驿站名；

D 选项 `len(stations)` 返回整数 5，同样不能迭代。

结论：应填入 `stations`。

5. 本题正确答案为 A. 需要 15 个箱子，最后一个箱子装 8 件

详细解析：

`total_porcelain=120`，`box_capacity=8`。整数除法得 `num_boxes=15`，取余得 `last_box=0`。

因为 `last_box==0`，执行 `else` 分支将 `last_box` 改为 8，`num_boxes` 不变仍为 15。输出 15 个箱子，最后一个装 8 件。

错误选项 B（15 箱、最后 0 件）忽略了 `else` 分支的修正；

C 和 D 的 16 箱均错误（无需加 1）。

结论：正确输出为 A。

6. 本题正确答案为 C. 函数必须有一个 `return` 语句

详细解析：

A 选项正确，函数使用 `return` 语句返回值；

B 选项正确，没有 `return` 语句时函数返回 `None`；

C 选项错误，函数不一定必须有 `return` 语句，可以没有（此时隐式返回 `None`）；

D 选项正确，`return` 后面可以跟表达式（如 `return a+b`）。

结论：错误的是 C。

7. 本题正确答案为 B. 使用一个字典（`dict`），键为城市名，值为数量

详细解析：

题目需要记录城市与抵达数量的对应关系。

A 选项用 5 个独立变量，管理繁琐且无法直接通过城市名查找；

B 选项字典通过键值对存储，既保留城市名又能快速访问数量，最合适；

C 选项列表只能存数量，丢失城市名信息；

D 选项集合无序且不能存储键值对。

结论：应使用字典。

8. 本题正确答案为 C. `caravan_size[2]`

详细解析：

Python 列表索引从 0 开始，第一个元素索引 0，第二个索引 1，第三个索引

2。

列表 `caravan_size = [5, 8, 12, 6, 10]` 中，第三个商队人数 12 对应索引 2。

A 选项索引 0 得 5，B 选项索引 1 得 8，D 选项索引 3 得 6。

结论：正确方式为 `caravan_size[2]`。

9. 本题正确答案为 B. 1415

详细解析：

代码通过 `for` 循环遍历索引，将每种商品的单价与对应数量相乘并累加到 `total_value`。

计算过程： $30 \times 10 = 300$ ， $15 \times 20 = 300$ ， $80 \times 3 = 240$ ， $25 \times 15 = 375$ ， $40 \times 5 = 200$ ，总和为 $300 + 300 + 240 + 375 + 200 = 1415$ 。

结论：总价值为 1415。

10. 本题正确答案为 C. `total += num`

详细解析：

`for` 循环遍历列表 `caravans`，每次迭代将当前元素赋值给 `num`。累加总和需用 `total += num`，将每个商队数量累加。

A 选项中的 `i` 未定义；

B 选项中的 `i` 同样未定义；

D 选项 `total = num` 会覆盖之前的累加值，导致只保存最后一个数。

结论：正确选项为 C。

11. 本题正确答案为 B. `import math`

详细解析：

Python 中使用数学函数 `sqrt()` 需先导入 `math` 模块。正确语法是 `import math`，然后通过 `math.sqrt()` 调用。

A 选项 `include math` 是 C/C++ 语法；

C 选项 `using math` 是其他语言写法；

D 选项 `#include <cmath>` 是 C++ 头文件包含。

结论：应使用 `import math`。

12. 本题正确答案为 D. `camel_count`

详细解析：

Python 变量命名规范要求：不能以数字开头，不能包含连字符（会被解析为减号），一般推荐使用小写字母加下划线的 `snake_case` 风格。

A 选项以数字 1 开头，非法；

B 选项包含短横线，非法；

C 选项 `camelCount` 为驼峰命名，虽合法但非 Python 最推荐风格；

D 选项 `camel_count` 符合 `snake_case` 规范，清晰且最适用于表示“骆驼数量”。

结论：应选 D。

13. 本题正确答案为 D. `bool`

详细解析：

题目中通行证状态只有“有效”或“无效”两种，属于布尔类型。Python 中 `bool` 类型值为 `True` 或 `False`，语义清晰且适合逻辑判断。

A 选项 `int` 虽可用 1/0 表示，但不够直观；

B 选项 `float` 用于小数；

C 选项 `str` 用于字符串。

结论：应使用 `bool`。

14. 本题正确答案为 A. 香料：30 袋，丝绸：0 匹

详细解析：

代码先计算 $\text{spice} = \text{silk} * \text{exchange_rate} = 10 \times 3 = 30$ ，然后将 silk 赋值为 0。最终输出香料 30 袋，丝绸 0 匹。

错误选项 B 误将 spice 当作 silk 的值；C 未执行 $\text{silk}=0$ ；D 计算错误。

结论：应选 A。

15. 本题正确答案为 B. $\text{total_tax} = (\text{price} + \text{freight}) * \text{tax_rate}$

详细解析：

公式为总税 = (价格 + 运费) × 税率。

A 选项先乘后加，相当于 $\text{price} + (\text{freight} \times \text{tax_rate})$ ，结果错误；

B 选项正确使用括号先求和再乘税率，得到浮点数 12.0；

C 选项使用 $\text{int}()$ 强转，会丢失小数部分（虽本例为整数但失去浮点性）；

D 选项虽结果正确但显式转换多余。题目要求“精确的浮点数结果”，B 最简洁恰当。

结论：应选 B。

16. (×)

解析：在 Python 中，= 是赋值运算符，用于将右边的值赋给左边变量；判断两个值是否相等应使用关系运算符 ==。因此说法错误。

17. (×)

解析：列表 trade_route 长度为 3，元素索引范围为 0 到 2，最后一个元素索引为 2（即 $\text{trade_route}[2]$ ），索引 3 会导致 IndexError 。因此说法错误。

18. (×)

解析：while 循环在执行前先判断条件，若条件为假则循环体一次也不执行。因此“至少执行一次”是错误的，只有 do...while（Python 无此结构）才至少执行一次。

19. (√)

解析：Python 中函数参数传递的是引用。如果形参是可变对象（如列表、字典），在函数内部修改该对象的内容（如添加、删除元素或修改元素值），会直接影响外部的实参。

20. (√)

解析：字符串支持索引操作， $\text{city}[0]$ 返回字符串的第一个字符。city = 'Dunhuang' 中第一个字符为 'D'，因此说法正确。