

全国青少年信息素养大赛-2026-小高组-算法-Py-丝路-复赛-模拟题-01-答案与解析

1	2	3	4	5	6	7	8	9	10
A	C	A	B	C	A,B,C	A,B,D	B,C	A,C,D	A,B,C, D

一、单选题（每题 4 分，共 20 分）

(1) 丝绸商队要将 17 匹丝绸平分给 5 个行商队。执行以下代码后，每队分到几匹、剩几匹、精确值是多少？（ ）

```
1 silk = 17
2 teams = 5
3 print(silk // teams, silk % teams)
4 print(silk / teams)
```

- A. 3 2 然后 3.4
- B. 3.4 2 然后 3.4
- C. 3 3 然后 3.4
- D. 3 2 然后 3

【正确答案】

A

【题目解析】

✔ 正确答案

A (3 2 然后 3.4)

💡 核心考点

整除 `//`、取余 `%` 与真除 `/` 的区别。

🔍 执行追踪

$17//5 = 3$, $17\%5 = 2$, $17/5 = 3.4$ 。

⚠ 易错提醒

/ 总是返回浮点数，// 才是向下取整。

(2) 贸易站清点了一批货物重量数据，需要按从重到轻排序后找出第二重的货物。执行以下代码后，输出结果为（ ）。

```
1 cargo = [850, 420, 960, 310, 1200]
2 cargo.sort(reverse=True)
3 print(cargo[1])
```

A. 310

B. 420

C. 960

D. 1200

【正确答案】

C

【题目解析】

✅ 正确答案

C (960)

💡 核心考点

列表降序排序与索引访问。

🔍 执行追踪

排序后列表为 [1200, 960, 850, 420, 310]，cargo[1] 即第二个元素 960。

⚠ 易错提醒

索引从 0 开始，cargo[1] 是第二重而非最重。

(3) 商队翻译官截获了一串丝路密语 'ABCDEFGH'，需要分别用"倒序读取"和"跳位读取"两种方式解码。执行以下代码后，输出结果为（ ）。

```
1 s = 'ABCDEFGH'
2 print(s[::-1])
3 print(s[::2])
```

- A. GFEDCBA 和 ACEG
- B. G 和 ACEG
- C. GFEDCBA 和 ABCDEFG
- D. ABCDEFG 和 ACEG

【正确答案】

A

【题目解析】

✅ 正确答案

A (GFEDCBA 和 ACEG)

💡 核心考点

字符串切片 `[::step]` 的步长用法。

🔍 执行追踪

`s[::-1]` 步长为 `-1`，倒序输出 `GFEDCBA`；`s[::2]` 步长为 `2`，从索引 `0` 起隔位取字符，得 `ACEG`。

⚠️ 易错提醒

负步长表示反向读取，不是只取最后一个字符。

(4) 丝路物流公司派出 3 个运输组，每组连续完成 4 趟货运任务。执行以下代码后，总共完成了多少次货运任务？ ()

```
1 count = 0
2 for i in range(3):
3     for j in range(4):
4         count += 1
5 print(count)
```

- A. 7
- B. 12
- C. 3
- D. 4

【正确答案】

B

【题目解析】

✓ 正确答案

B (12)

💡 核心考点

双重 `for` 循环次数计算。

🔍 执行追踪

外层执行 3 次，内层执行 4 次，共 $3 \times 4 = 12$ 次。

⚠ 易错提醒

嵌套循环总次数是相乘而非相加。

(5) 关市管理员尝试将外商的货物编号字符串转换为数字以便登记。如果编号中包含非数字字符，转换将失败。执行以下代码后，输出结果为 ()。

```
1 try:
2     a = int('X12B')
3     print(a)
4 except ValueError:
5     print('编号格式错误')
```

- A. X12B
- B. 0
- C. 编号格式错误
- D. 程序报错并终止

【正确答案】

C

【题目解析】

✓ 正确答案

C (编号格式错误)

💡 核心考点

考察 `try-except` 异常捕获机制。

🔍 执行追踪

`int('X12B')` 抛出 `ValueError`，被 `except` 捕获，执行 `print('编号格式错误')`。

⚠ 易错提醒

异常被捕获后程序不会终止，会继续执行 `except` 块中的语句。

二、多选题（每题 4 分，共 20 分；少选得 2 分，错选不得分）

(6) 贸易站用函数格式化通关文牒内容。执行以下代码，结果正确的有（ ）。

```
1 def visa(name, goods='丝绸'):  
2     return f'{name}—携带{goods}通关'  
3  
4 a = visa('阿里')  
5 b = visa('马克', '茶叶')  
6 c = visa(goods='瓷器', name='哈桑')
```

- A. `a` 的值是 `'阿里—携带丝绸通关'`
- B. `b` 的值是 `'马克—携带茶叶通关'`
- C. `c` 的值是 `'哈桑—携带瓷器通关'`
- D. `visa('郑和', goods='香料')` 会报错（位置参数不能在关键字参数之后）

【正确答案】

A,B,C

【题目解析】

✅ 正确答案

A、B、C

💡 核心考点

函数默认参数、关键字参数调用规则。

🔍 执行追踪

A 使用默认值得到 `'阿里—携带丝绸通关'`；B 覆盖默认值得到 `'马克—携带茶叶通关'`；C 使用关键字传参得到 `'哈桑—携带瓷器通关'`；D 中位置参数 `'郑和'` 在关键字参数之前，合法不报错。

⚠ 易错提醒

只要位置参数在关键字参数之前就合法，D 项描述错误。

(7) 关于贸易清单（列表）和固定航线表（元组），以下说法正确的有（ ）。

- A. 列表是可变的，可以修改、增加、删除元素
- B. 元组是不可变的，创建后不能修改其中的元素
- C. 元组可以使用 `append()` 方法添加元素
- D. 列表和元组都支持索引和切片操作

【正确答案】

A,B,D

【题目解析】

✅ 正确答案

A、B、D

💡 核心考点

列表与元组的特性区别。

🔍 执行追踪

列表 (list) 可变，支持增删改；元组 (tuple) 不可变，创建后不能修改；两者都是序列类型，均支持索引和切片。

⚠️ 易错提醒

元组没有 `append()` 方法，只有列表才有，故 C 错误。

(8) 丝路贸易档案管理中，已知 `records = {'丝绸': 101, '茶叶': 202}`，以下操作会导致错误的有 ()。

- A. `records['瓷器'] = 303`
- B. `records[['香料']] = 'a'`
- C. `print(records['珠宝'])`
- D. `records['丝绸'] = 1001`

【正确答案】

B,C

【题目解析】

✅ 正确答案

B、C

💡 核心考点

字典的键类型限制与键不存在的访问异常。

🔍 执行追踪

A 新增键值对合法；B 列表 `['香料']` 不可哈希，报 `TypeError`；C 键 `'珠宝'` 不存在，报 `KeyError`；D 修改已有键值合法。

⚠️ 易错提醒

字典的键必须是不可变（可哈希）类型，列表不能作为键；访问不存在的键会抛出 `KeyError`，需用 `get()` 避免。

(9) 关于各商队内部使用的货物统计函数（作用域），以下说法正确的有（ ）。

- A. 函数内部定义的变量默认是局部变量，函数外部无法访问
- B. 在函数外部定义的变量在函数内部可以直接修改
- C. 使用 `global` 关键字可以在函数内部修改全局变量
- D. 不同函数内部可以有同名的局部变量，互不影响

【正确答案】

A,C,D

【题目解析】

✅ 正确答案

A、C、D

💡 核心考点

Python 函数作用域（LEGB 规则）与 `global` 关键字。

🔍 执行追踪

A 正确，函数内变量默认为局部变量；B 错误，函数外的变量在函数内只能读取，不能直接赋值修改（赋值会创建新的局部变量）；C 正确，`global` 声明后可在函数内修改全局变量；D 正确，不同函数的局部变量作用域相互独立。

⚠️ 易错提醒

注意区分“访问”和“修改”——不使用 `global` 时，函数内可以读取全局变量的值，但赋值操作会被解释为创建同名局部变量。

(10) 丝路贸易协会在评估新航线开通的可行性，运行以下条件判断代码：

```
1 x, y = 10, 3
2 res1 = x > 5 and y < 5
3 res2 = not (x == y)
4 res3 = x % y == 1 and x // y == 3
```

结果正确的有（ ）。

- A. `res1` 的值是 `True`
- B. `res2` 的值是 `True`
- C. `res3` 的值是 `True`
- D. `x >= 10 or y > 5` 的值是 `True`

【正确答案】

A,B,C,D

【题目解析】

✅ 正确答案

A、B、C、D

💡 核心考点

考察布尔运算符 `and`、`or`、`not` 与算术运算符的组合判断。

🔍 执行追踪

`res1` : $10 > 5$ 且 $3 < 5$, 为 `True` ; `res2` : $10 \neq 3$, `not False` 为 `True` ; `res3` : $10 \% 3 = 1$ 且 $10 // 3 = 3$, 为 `True` ; D 项 $x \geq 10$ 成立, `or` 短路为 `True` 。

⚠️ 易错提醒

注意 `and` 要求两边同真, `or` 只需一边真; `//` 是整除, `%` 是取余, 别混淆。

三、编程题（每题 15 分，共 60 分）

(11) S2513 驿站住宿费用累计

【题目描述】

丝路商旅按以下规则在各驿站支付住宿费：

- 第 1 天，每人付 1 个银币；
- 第 2 天和第 3 天，每人每天付 2 个银币；

- 第 4、5、6 天，每人每天付 3 个银币；
- 第 7、8、9、10 天，每人每天付 4 个银币.....
- 以此类推：当连续 N 天每人每天付 N 个银币后，接下来的连续 $N + 1$ 天里，每人每天付 $N + 1$ 个银币。

请你编写程序：输入总天数 K ，计算前 K 天里每人一共累计支付了多少个银币。

【输入描述】

一个正整数 K （ $1 \leq K \leq 10000$ ），表示住宿天数。

【输出描述】

一个正整数，表示每人累计支付的银币总数。

【输入样例 1】

1 6

【输入样例 2】

1 1000

【输出样例 1】

1 14

【输出样例 2】

1 29820

【答案解析】

💡 解题思路

按阶段累加费用。第 i 个阶段连续 i 天，每天付 i 个银币，该阶段共付 $i \times i$ 个银币。用循环逐阶段处理，直到剩余天数不足一个完整阶段。

🔍 逻辑推演

- 天数 1：第 1 阶段（1 天，每天 1 银币） $\rightarrow 1 \times 1 = 1$
- 天数 2~3：第 2 阶段（2 天，每天 2 银币） $\rightarrow 2 \times 2 = 4$
- 天数 4~6：第 3 阶段（3 天，每天 3 银币） $\rightarrow 3 \times 3 = 9$
- 前 6 天累计： $1 + 4 + 9 = 14$

⚠️ 注意

最后一个阶段可能不完整（剩余天数不足 $N + 1$ 天），此时按实际剩余天数计算。

```
1 k = int(input())
2 total = 0
3 n = 1 # 当前阶段编号，表示每天付 n 银币
4 while k > 0:
5     days = min(n, k) # 本阶段实际天数
6     total += days * n
7     k -= days
8     n += 1
9 print(total)
```

(12) S2514 贸易单据编号校验

【题目描述】

每一批丝路贸易货物都有一个标准编号，格式为： `x-xxx-xxxxx-x`，其中符号 `-` 是分隔符，最后一位是校验码。编号共包含 9 位数字、1 位校验码和 3 个分隔符。

校验码的计算方法如下：将编号中的 9 位数字（忽略分隔符）从左到右依次乘以 $1, 2, 3, \dots, 9$ ，求和后用结果对 11 取余。如果余数为 10，则校验码为大写字母 `X`；否则校验码为余数本身。

请你编写程序：输入一个货物编号，判断其校验码是否正确。如果正确，输出 `Right`；如果错误，输出修正后的正确编号。

【输入描述】

一个字符串，表示货物编号（保证格式为 `x-xxx-xxxxx-x`）。

【输出描述】

如果校验码正确，输出 `Right`；否则输出正确的完整编号（含分隔符 `-`）。

【输入样例 1】

```
1 0-670-82162-4
```

【输入样例 2】

```
1 0-670-82162-0
```

【输出样例 1】

```
1 Right
```

【输出样例 2】

1 0-670-82162-4

【答案解析】

💡 解题思路

1. 提取编号中的 9 位数字，去掉 - 分隔符
2. 计算加权 and：第 i 位数字 $\times i$ (i 从 1 到 9)
3. 对 11 取余得到正确校验码
4. 与原校验码比较

🔍 逻辑推演

以 0-670-82162-4 为例：

- 提取数字：067082162

- 加权 and：

$$0 \times 1 + 6 \times 2 + 7 \times 3 + 0 \times 4 + 8 \times 5 + 2 \times 6 + 1 \times 7 + 6 \times 8 + 2 \times 9 = 0 + 12 + 21 + 0 + 40 + 12 + 7 + 48 + 18 = 158$$

- $158 \% 11 = 4$ ，与校验码 4 一致

⚠️ 注意

余数为 10 时校验码是字母 X，不是数字 10。

```

1  code = input().strip()
2  # 提取前9位数字 (去掉所有-)
3  digits = code.replace('-', '')
4  nine_digits = digits[:9]
5  check_char = digits[9] # 原校验码
6
7  # 计算加权和
8  s = 0
9  for i in range(9):
10     s += int(nine_digits[i]) * (i + 1)
11
12  remainder = s % 11
13  if remainder == 10:
14     correct_check = 'X'
15  else:
16     correct_check = str(remainder)
17
18  if check_char == correct_check:
19     print('Right')
20  else:
21     # 输出修正后的编号
22     print(f'{nine_digits[0]}-{nine_digits[1:4]}-{nine_digits[4:9]}-{correct_check}')

```

(13) S2515 丝路商品价格连续下跌监测

【题目描述】

贸易协会记录了丝路某口岸连续 n 个月的香料交易价格。由于商路竞争激烈，市场行情持续波动，如果某月的价格比前一个月更低，则记为“下跌”。

请你编写程序，找出这 n 个月中最长的一段连续下跌的月数（即连续多个月，每个月的价格都比前一个月低），帮助管理部门判断该商品贸易是否需要调整策略。

【输入描述】

第一行，一个整数 n （ $1 \leq n \leq 100$ ），表示统计的月数。

第二行， n 个整数（空格分隔），表示每月的价格。

【输出描述】

一行，一个整数，表示最长连续下跌的月数。

【输入样例 1】

```
1 9
2 68 65 62 64 59 56 53 58 52
```

【输出样例 1】

```
1 4
```

【答案解析】

💡 解题思路

遍历价格数据，用变量记录当前连续下跌长度和最大连续下跌长度。当某月不低于前一月时，当前连续计数重置为 0。

🔍 逻辑推演

以 68 65 62 64 59 56 53 58 52 为例：

- 68→65→62：连续下跌 2 段（3 个月）
- 62→64：上升，重置
- 64→59→56→53：连续下跌 3 段（4 个月）
- 53→58：上升，重置
- 58→52：连续下跌 1 段（2 个月）
- 最大连续下跌月数为 4

⚠ 注意

连续下跌的“月数”是指包含首月在内的总月数。如果连续下跌 k 段，则月数为 $k + 1$ 。

```
1 n = int(input())
2 prices = list(map(int, input().split()))
3
4 if n == 0:
5     print(0)
6 else:
7     max_len = 1
8     cur_len = 1
9     for i in range(1, n):
10        if prices[i] < prices[i-1]:
11            cur_len += 1
12            max_len = max(max_len, cur_len)
13        else:
14            cur_len = 1
15    print(max_len)
```

(14) S2516 丝路贸易站点资源平衡

【题目描述】

丝路商贸委员会正在进行一项"丝路贸易站点资源平衡"计划。某条贸易路线上有 n 个贸易站点（自左至右编号 $1 \sim n$ ），相邻编号的站点之间相隔 1 千米。第 i 号站点里有 c_i 名贸易商。

陆路运输队位于左侧，代表"驼队"；海运船队位于右侧，代表"船队"。他们以 m 号站点作为分界，靠左的贸易商属于驼队势力，靠右的贸易商属于船队势力，而第 m 号站点中的贸易商不属于任何一方。

一个站点的影响力为：该站点中的贸易商人数 $\times$ 该站点到 m 号站点的距离；参与分配一方的实力定义为：属于这一方所有站点的影响力之和。

平衡过程中，突然有 s_1 位新贸易商加入了 p_1 号站点。作为委员会调度员，你知道如果双方实力差距太悬殊，合作就无法继续。为了让合作继续，你需要选择一个站点 p_2 ，并将你手里的 s_2 位新贸易商全部派往站点 p_2 ，使得双方实力差距尽可能小。

注意：你手中的新贸易商落在哪个站点，就和该站点中其他贸易商有相同的团队归属（如果落在 m 号站点，则不属于任何团队）。

【输入描述】

第一行，一个正整数 n ，代表站点的数量。

第二行， n 个正整数，相邻两数之间以一个空格分隔，第 i 个正整数代表编号为 i 的站点中起始时的贸易商数量 c_i 。

第三行，四个正整数 m 、 p_1 、 s_1 、 s_2 ，相邻两数间以一个空格分隔。

【输出描述】

一行，一个正整数 p_2 ，表示你选择的新贸易商投放站点编号。如果存在多个编号同时满足最优，取最小的编号。

【输入样例 1】

```
1 6
2 2 3 2 3 2 3
3 4 6 5 2
```

【输入样例 2】

```
1 6
2 1 1 1 1 1 16
3 5 4 1 1
```

【输出样例 1】

1 2

【输出样例 2】

1 1

【答案解析】

💡 解题思路

1. 先将 s_1 位贸易商加到 p_1 号站点
2. 枚举每个可能的 p_2 (1 到 n), 将 s_2 加到 p_2 号站点
3. 计算每种情况下的双方实力差的绝对值
4. 找出使差距最小的 p_2 , 多解时取编号最小

🔍 逻辑推演

实力计算公式:

- 驼队 (左侧) : $\sum_{i=1}^{m-1} c_i \times (m - i)$
- 船队 (右侧) : $\sum_{i=m+1}^n c_i \times (i - m)$

枚举 p_2 时, 需要考虑 s_2 加入后对哪一方产生影响:

- 若 $p_2 < m$, 驼队增加 $s_2 \times (m - p_2)$
- 若 $p_2 > m$, 船队增加 $s_2 \times (p_2 - m)$
- 若 $p_2 = m$, 不影响任何一方

⚠ 注意

多解时取编号最小的 p_2 。

```

1  n = int(input())
2  c = list(map(int, input().split()))
3  m, p1, s1, s2 = map(int, input().split())
4
5  # 将 s1 加入 p1 号站点
6  c[p1 - 1] += s1
7
8  best_p2 = 1
9  min_diff = float('inf')
10
11 for p2 in range(1, n + 1):
12     # 复制当前状态并加入 s2
13     temp_c = c.copy()
14     temp_c[p2 - 1] += s2
15
16     # 计算驼队实力 (左侧)
17     left_power = 0
18     for i in range(1, m):
19         left_power += temp_c[i - 1] * (m - i)
20
21     # 计算船队实力 (右侧)
22     right_power = 0
23     for i in range(m + 1, n + 1):
24         right_power += temp_c[i - 1] * (i - m)
25
26     diff = abs(left_power - right_power)
27     if diff < min_diff:
28         min_diff = diff
29         best_p2 = p2
30
31 print(best_p2)

```