

**GESP****CCF 编程能力等级认证**  
Grade Examination of Software Programming

# Python 四级

2025 年 12 月

## 1 单选题（每题 2 分，共 30 分）

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	B	B	B	C	B	A	B	A	C	C	D	B	C	A	C

**第1题** 近日，空中客车公司表示，约6000架空客A320系列飞机需要紧急更换一种易受太阳辐射影响的飞行控制软件。空客表示，在对一起飞行事故进行分析后，表明强烈的太阳辐射可能会损坏飞行控制系统所需的关键数据，导致判断失误，进而可能引发飞行异常。在这里的飞行控制系统中，执行判断的部件最可能是下面的( )。

- A. 辐射传感器
- B. 处理器
- C. 内存单元
- D. 输出设备

**第2题** 小杨最近为了备考GESP，开始看网络上关于网络知识的视频。其中提到计算机网络系统有不同的划分标准，其中之一就是按照地理覆盖的范围来划分。那他平时上学所在的教学楼内的网络是一个( )。

- A. PAN
- B. LAN
- C. MAN
- D. WAN

**第3题** 对于列表 `my_list = [['a', 'b'], 1, 2]`，执行哪项操作可以生成新列表 `['a', 'b', 'c']`？

- A. `new_list = my_list[0].append('c')`
- B. `new_list = my_list[0] + ['c']`
- C. `new_list = my_list[0].extend(['c'])`
- D. `new_list = my_list.pop(0).append('c')`

**第4题** 执行以下Python代码后，其输出是？( )

```
1 a = [1, 2, [3, 4]]
2 b = a[:]
3 b[0] = 10
4 b[2][0] = 30
5 print(a)
```

- A. `[1, 2, [3, 4]]`
- B. `[10, 2, [30, 4]]`
- C. `[1, 2, [30, 4]]`

D. [10, 2, [3, 4]]

第5题 执行以下Python代码后，输出的结果是? ( )

```
1 matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
2 result = [matrix[i][i] for i in range(len(matrix))]
3 print(result)
```

- A. [1, 2, 3]
- B. [1, 5, 9]
- C. [3, 5, 7]
- D. [1, 4, 7]

第6题 执行以下Python代码后，输出的结果是? ( )

```
1
2 def modify(s):
3     s = s.upper()
4     return s
5
6 text = "hello"
7 modify(text)
8 print(text)
```

- A. hello
- B. HELLO
- C. None
- D. 报错

第7题 执行下面Python代码时，哪条调用会报错? ( )

```
1 def func(*args, **kwargs):
2     print(args, kwargs)
```

- A. func(1, 2, x=3, y=4)
- B. func(a=1, b=2, 3, 4)
- C. func(1, 2, [3, 4])
- D. func(\*[1, 2], \*\*{'x': 3})

第8题 执行以下Python代码后，输出的结果是? ( )

```
1 def func(*args):
2     return sum(args)
3
4 print(func(1, 2, *[3, 4]))
```

- A. 10
- B. 6
- C. 7
- D. 报错

第9题 关于Python中全局变量与局部变量的描述，以下哪一项是错误的? ( )

- A. 局部变量在函数退出后失效
- B. 函数内需用 `global` 声明才能修改全局变量
- C. 全局变量不能和局部变量重名
- D. 使用 `nonlocal` 关键字可以在嵌套函数中修改其直接外层（非全局）作用域中的变量。

第 10 题 执行以下Python代码后，输出的结果是? ( )

```
1 with open('test.txt', 'w') as f:  
2     f.write('Hello')  
3  
4 with open('test.txt', 'a') as f:  
5     f.write('World')  
6  
7 with open('test.txt', 'r') as f:  
8     print(f.read())
```

- A. Hello
- B. World
- C. HelloWorld
- D. WorldHello

第 11 题 在Python中，以下哪个选项可以正确地读取一个二进制文件并打印其内容? ( )

- A. `with open('image.bmp', 'r') as f: print(f.read())`
- B. `with open('image.bmp', 'w') as f: print(f.read())`
- C. `with open('image.bmp', 'a') as f: print(f.read())`
- D. `with open('image.bmp', 'rb') as f: print(f.read())`

第 12 题 执行以下Python代码，如果文件 `data.txt` 不存在，输出结果是? ( )

```
1 try:  
2     with open('data.txt', 'r') as f:  
3         content = f.read()  
4         print("File read successfully.")  
5     except FileNotFoundError:  
6         print("File not found.")  
7     except Exception as e:  
8         print("An error occurred:", e)
```

- A. File read successfully.
- B. File not found.
- C. An error occurred: ...
- D. 没有任何输出

第 13 题 以下Python代码的时间复杂度是? ( )

```
1 def func(n):  
2     i = 0  
3     while i * i * i < n:  
4         i += 1
```

- A.  $O(n)$

- B.  $O(n \log n)$
- C.  $O(\sqrt[3]{n})$
- D.  $O(\sqrt{n})$

第14题 下面程序空白处填上哪些内容可以实现程序的由小到大排序功能? ( )

```

1 def selection_sort(arr):
2     n = len(arr)
3     for i in range(n-1):
4         min_index = i # 初始化最小索引
5         for j in range(i+1, n):
6             if arr[j] < arr[min_index]:
7                 ----- # 填写代码 ①
8                 ----- # 填写代码 ②

```

- A. ① `min_index = j` ② `arr[i], arr[min_index] = arr[min_index], arr[i]`
- B. ① `min_index = i` ② `arr[i], arr[j] = arr[j], arr[i]`
- C. ① `j = min_index` ② `arr[min_index], arr[i] = arr[i], arr[min_index]`
- D. ① `min_index = j` ② `arr[j], arr[i] = arr[i], arr[j]`

第15题 执行以下Python代码，输出结果是?

```

1 items = [("book", 20), ("pen", 5), ("bag", 30)]
2 sorted_items = sorted(items, key=lambda x: x[1], reverse=True)
3 print(sorted_items[0])

```

- A. ("book", 20)
- B. ("pen", 5)
- C. ("bag", 30)
- D. 30

## 2 判断题 (每题 2 分, 共 20 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	✗	✗	✗	✓	✓	✓	✗	✗	✓	✗

第1题 鸿蒙是华为公司开发的一款操作系统，那么它能够将正确的源程序翻译成目标程序，并运行。 ( )

第2题 执行以下Python代码后，输出结果是 `([1, 2], 3)`。

```

1 a = [1, 2]
2 b = (a, 3)
3 a.append(4)
4 print(b)

```

第3题 使用解析式 `{x: y for x in range(2) for y in range(2)}` 可以创建一个包含4个键值对的字典。

第4题 执行以下Python代码后，输出结果是 15。

```
1 | def create_multiplier(n):
2 |     return lambda x: x * n
3 |
4 | mult = create_multiplier(3)
5 | print(mult(5))
```

第 5 题 执行以下Python代码后，输出结果是 True。

```
1 | def func(lst=[]):
2 |     lst.append(1)
3 |     return lst
4 |
5 | print(func() is func())
```

第 6 题 在Python中，`readline()` 方法每次读取文件的下一行内容，包括行尾的换行符。如果已到文件末尾，则返回一个空字符串。

第 7 题 执行下面Python代码会抛出 `TypeError` 异常。

```
1 | int("3.14")
```

第 8 题 执行下面Python代码后，输出的结果为 [0]。

```
1 | print(list(filter(None, [0, 1, 2, 3])))
```

第 9 题 选择排序算法是不稳定的，而插入排序算法是稳定的。

第 10 题 以下程序中使用了递推方式计算阶乘 ( $n! = 1 \times 2 \times \dots \times n$ )，计算结果正确。

```
1 | def factorial(n):
2 |     res = 1
3 |     for i in range(n + 1):
4 |         res *= i
5 |
6 |     return res
```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：建造
- 时间限制：3.0 s
- 内存限制：512.0 MB

##### 3.1.1 题目描述

小A有一张  $M$  行  $N$  列的地形图，其中第  $i$  行第  $j$  列的数字  $a_{ij}$  代表坐标  $(i, j)$  的海拔高度。

停机坪为一个  $3 \times 3$  的区域且内部所有 9 个点的最大高度和最小高度之差不超过  $H$ 。

小A想请你计算出，在所有适合建造停机坪的区域中，区域内部 9 个点海拔之和最大是多少。

### 3.1.2 输入格式

第一行三个正整数  $M, N, H$ , 含义如题面所示。

之后  $M$  行, 第  $i$  行包含  $N$  个整数  $a_{i1}, a_{i2}, \dots, a_{iN}$ , 代表坐标  $(i, j)$  的高度。

数据保证总存在一个适合建造停机坪的区域。

### 3.1.3 输出格式

输出一行, 代表最大的海拔之和。

#### 3.1.4 样例

##### 3.1.4.1 输入样例

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

##### 3.1.4.2 输出样例

```
1 | 40
```

### 3.1.5 数据范围

对于所有测试点, 保证  $1 \leq M, N \leq 10^3, 1 \leq H, a_{ij} \leq 10^5$ 。

### 3.1.6 参考程序

```
1 # 获取长、宽、最大高度差
2 M, N, H = map(int, input().split())
3 # 构建嵌套列表, 这个列表包含M个长度为N的列表
4 a = [list(map(int, input().split())) for _ in range(M)]
5 # 海拔之和的最大值
6 max_sum = 0
7
8 # 遍历所有可能的 3x3 区域的左上角
9 for i in range(M - 2):
10     for j in range(N - 2):
11         # 提取 3x3 区域内的数值
12         # 方法: 拼接三行的切片
13         block = a[i][j:j+3] + a[i+1][j:j+3] + a[i+2][j:j+3]
14         # 判断这一区域最高最低点的海拔差
15         if max(block) - min(block) <= H:
16             max_sum = max(max_sum, sum(block))
17
18 print(max_sum)
```

## 3.2 编程题 2

- 试题名称: 优先购买
- 时间限制: 3.0 s
- 内存限制: 512.0 MB

### 3.2.1 题目描述

小A有 $M$ 元预算。商店有 $N$ 个商品，每个商品有商品名 $S$ 、价格 $P$ 和优先级 $V$ 三种属性，其中 $V$ 为正整数，且 $V$ 越小代表商品的优先级越高。

小A的购物策略为：

- 总是优先买优先级最高的东西；
- 如果有多个最高优先级商品，购买价格最低的；
- 如果有多个优先级最高且价格最低的商品，购买商品名字典序最小的。

小A想知道能购买哪些商品。

### 3.2.2 输入格式

第一行两个正整数 $M, N$ ，代表预算和商品数。

之后 $N$ 行，每行一个商品，依次为 $S_i P_i V_i$ ，代表第 $i$ 个商品的商品名、价格、优先级。

数据保证不存在两个名字相同的商品。

### 3.2.3 输出格式

按照字典序从小到大的顺序，输出所有购买商品的商品名。

### 3.2.4 样例

#### 3.2.4.1 输入样例

```
1 20 4
2 apple 6 8
3 bus 15 1
4 cab 1 10
5 water 4 8
```

#### 3.2.4.2 输出样例

```
1 bus
2 cab
3 water
```

### 3.2.5 数据范围

对于所有测试点，保证 $1 \leq |S_i| \leq 10, 1 \leq M, P_i \leq 10^5, 1 \leq N \leq 10^3, 1 \leq V_i \leq 10$ 。商品名仅由小写字母组成且不存在两个相同的商品名。

### 3.2.6 参考程序

```
1 # 预算M与商品数量N
2 M, N = map(int, input().split())
3
4 # 各个商品组成的列表，每一个商品是一个tuple，第一项是名称，第二项是价格，第三项是优先级
5 items = []
6 for _ in range(N):
7     # 读入 name, price, priority
8     data = input().split()
9     items.append((data[0], int(data[1]), int(data[2])))
10
11 # 第一次排序：优先级(x[2]) -> 价格(x[1]) -> 名字(x[0])
12 # 优先级数值较低（也就是优先级更高）在前
13 # 优先级相等则价格更低在前
14 # 价格相等则按照字典序从小到大排序
15 items.sort(key=lambda x: (x[2], x[1], x[0]))
16
17 # 遍历列表，在预算范围内选择商品
18 ans = []
19 for name, price, _ in items:
20     if price <= M:
21         M -= price
22         ans.append(name)
23
24 # 第二次排序：按名字字典序，输出商品名称
25 ans.sort()
26
27 for name in ans:
28     print(name)
```