



Python 四级

2025 年 09 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 | D | C | B | B | C | B | A | C | D | D | B | C | C | B | A |

第 1 题 人工智能现在非常火，小杨就想多了解一下，其中就经常听人提到“大模型”。那么请问这里说的“大模型”最贴切是指（ ）。

- ☐ A. 大电脑模型
- ☐ B. 大规模智能
- ☐ C. 智能的单位
- ☐ D. 大语言模型

第 2 题 在TCP协议中，完成连接建立需要通过（ ）握手。

- ☐ A. 一次
- ☐ B. 二次
- ☐ C. 三次
- ☐ D. 四次

第 3 题 执行下面Python代码后，输出的结果是？（ ）

```
1 t = (1, [2, 3], 4)
2 t[1].extend([5, 6])
3 print(t)
```

- ☐ A. (1, [2, 3], 4)
- ☐ B. (1, [2, 3, 5, 6], 4)
- ☐ C. (1, [2, 3, [5, 6]], 4)
- ☐ D. 报错

第 4 题 执行下面Python代码后，输出的结果是？（ ）

```
1 data = [('a', [1]), ('b', [2])]
2 d = dict(data)
3 d['a'].append(3)
4 print(data[0][1])
```

- ☐ A. [1]
- ☐ B. [1, 3]
- ☐ C. 报错

☐ D. ('a', [1, 3])

第5题 以下哪种函数定义是正确的？（）

☐ A. def func(a=1, b, *args): pass

☐ B. def func(a, b=2, **kwargs, *args): pass

☐ C. def func(a, *args, b=2, **kwargs): pass

☐ D. def func(**kwargs, *args, a): pass

第6题 执行下面Python代码后，输出的结果是？（）

```
1 def func(a, lst=[]):
2     lst.append(a)
3     return lst
4
5 print(func(1), end=" ")
6 print(func(2))
```

☐ A. [1] [2]

☐ B. [1] [1, 2]

☐ C. [1] [2, 1]

☐ D. 报错

第7题 执行下面Python代码后，输出的结果是？（）

```
1 def demo(x, y, z):
2     return x + y + z
3
4 values = (1, 2, 3)
5 result = demo(*values)
6 print(result)
```

☐ A. 6

☐ B. (1, 2, 3)

☐ C. 报错

☐ D. 123

第8题 下列有关函数内修改全局变量描述正确的是？（）

☐ A. 函数内部可以直接修改任何全局变量

☐ B. 函数内部无法修改全局变量，只能读取

☐ C. 需要使用 global 关键字声明后才能修改全局变量

☐ D. 需要使用 nonlocal 关键字声明后才能修改全局变量

第9题 下列有关Python处理异常的说法正确的是？（）

☐ A. 每一个 try 模块只能设定一个 except 模块。

☐ B. finally 模块仅在 try 模块中没有异常发生时才会执行。

☐ C. else 模块会在 except 模块之后，finally 模块之前执行。

☐ D. 异常处理可以弥补程序漏洞，使得程序在一些情况下不会终止运行。

第10题 为了安全地打开并读取一个文件，确保即使发生异常，文件也能被正确关闭，最佳做法是？（）

- ☐ A. 将 `file.close()` 放在 `try` 块的最后一行。
- ☐ B. 将 `file.close()` 放在 `except` 块中。
- ☐ C. 将 `file.close()` 放在 `else` 块中。
- ☐ D. 使用 `with open('file.txt') as f:` 语句。

第 11 题 文件 `votes.txt` 内容如下：

```
1 Alice
2 Bob
3 Alice
4 Charlie
5 Bob
6 Alice
7 Alice
```

执行下面Python代码后，输出的结果是？（ ）

```
1 def count_votes(filename):
2     vote_count = {}
3     file = None
4     try:
5         file = open(filename, 'r')
6         for line in file:
7             name = line.strip()
8             if name in vote_count:
9                 vote_count[name] += 1
10            else:
11                vote_count[name] = 1
12    except FileNotFoundError:
13        print(f"错误: 文件 {filename} 不存在。")
14    except IOError:
15        print(f"错误: 读取文件 {filename} 时发生I/O错误。")
16    finally:
17        if file:
18            file.close()
19    return vote_count
20
21
22 result = count_votes('votes.txt')
23 print(result)
```

- ☐ A. `['Alice', 'Bob', 'Alice', 'Charlie', 'Bob', 'Alice', 'Alice']`
- ☐ B. `{'Alice': 4, 'Bob': 2, 'Charlie': 1}`
- ☐ C. `{4, 2, 1}`
- ☐ D. `{'Alice', 'Bob', 'Charlie'}`

第 12 题 以下代码的时间复杂度是？（ ）

```
1 def process_data(n):
2     count = 0
3     for i in range(n):
4         for j in range(n):
5             count += i * j
6     return count
```

- ☐ A. $O(1)$
- ☐ B. $O(n)$

- ☐ C. $O(n^2)$
- ☐ D. $O(\log n)$

第 13 题 下面代码采用递推算算法来计算斐波那契数列 $f(n) = f(n-1) + f(n-2)$ ，则横线上应填写？（）

```
1 def fibo(n):
2     if n <= 0:
3         return 0
4     elif n == 1:
5         return 1
6     else:
7         a, b = 0, 1
8         for i in range(2, n + 1):
9             _____ # 划线处
10        return b
```

- ☐ A. `a = b + a`
- ☐ B. `a, b = a + b, a`
- ☐ C. `a, b = b, a + b`
- ☐ D. `b = a + b`

第 14 题 以下是Python冒泡排序（由小到大）的优化版本，加入了提前终止机制。划线处应填入？（）

```
1 def optimized_bubble_sort(arr):
2     n = len(arr)
3     for i in range(n - 1, 0, -1):
4         swapped = _____ # 初始化交换标志
5         for j in range(0, i):
6             if arr[j] > arr[j + 1]:
7                 arr[j], arr[j + 1] = arr[j + 1], arr[j]
8                 swapped = _____ # 划线处
9         if not swapped:
10            break
```

- ☐ A. `True False`
- ☐ B. `False True`
- ☐ C. `True break`
- ☐ D. `False continue`

第 15 题 执行下面Python代码后，输出的结果是？（）

```
1 words = ["Apple", "banana", "cherry", "Date"]
2 sorted_words = sorted(words, key=str.lower, reverse=True)
3 print(sorted_words)
```

- ☐ A. `['Date', 'cherry', 'banana', 'Apple']`
- ☐ B. `['Apple', 'banana', 'cherry', 'Date']`
- ☐ C. `['date', 'cherry', 'banana', 'apple']`
- ☐ D. `['apple', 'banana', 'cherry', 'date']`

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 在集成开发环境里调试程序时，要注意不能修改源程序，因为如果修改，就要终止调试、关闭该文件并重新打开，才能再次开始调试。（ ）

第 2 题 执行下面Python代码后，输出的结果为 qu 。

```
1 data = {'numbers': [10, 20, 30], 'title': 'quiz'}
2 data["title"] += str(data['numbers'].pop(-2))
3 print(data['title'][:-2])
```

第 3 题 执行下面代码后，输出的结果为 5 。

```
1 d = {'a': 3, 'b': 5, 'c': 1}
2 print(max(d, key=lambda k: d[k]))
```

第 4 题 在 Python 中，如果一个函数没有执行到任何 return 语句，或者 return 语句后没有指定返回值，Python 解释器会默认让函数返回 None 。

第 5 题 执行下面Python代码会抛出 ValueError 异常。

```
1 len(123)
```

第 6 题 若文件log.txt已存在，执行以下代码后文件内容被清空，仅保留 New log entry 。

```
1 with open("log.txt", "w") as f:
2     f.write("New log entry")
```

第 7 题 以下代码的空间复杂度是 $O(n)$ 。

```
1 def create_dict(n):
2     my_dict = {}
3     for i in range(n):
4         my_dict[i] = i * 2
5     return my_dict
```

第 8 题 递推算法必须存在至少一个初始值，这是递推算法能够启动和正确运行的基础。

第 9 题 选择排序和插入排序的平均时间复杂度都是 $O(n^2)$ ，因此它们的效率在任何情况下都完全相同。

第 10 题 你收集了一些咖啡馆的信息，存储在字典 coffee_shops 中，以下代码可以筛选出所有评分（rating）高于 4.5 且距离（distance）在 1 公里以内的咖啡馆。

```
1 coffee_shops = [
2     {"name": "A Cafe", "rating": 4.7, "distance": 1.2},
3     {"name": "B Brew", "rating": 4.3, "distance": 0.5},
4     {"name": "C Coffee", "rating": 4.9, "distance": 2.1},
5     {"name": "D Delight", "rating": 4.6, "distance": 0.9}
6 ]
7
8 filtered_shops = list(
9     filter(lambda shop: shop["rating"] > 4.5 and shop["distance"] < 1.0, coffee_shops)
10 )
11 print(filtered_shops)
```

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称：排兵布阵
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.1.1 题目描述

作为将军，你自然需要合理地排兵布阵。地图可以视为 n 行 m 列的网格，适合排兵的网格以 1 标注，不适合排兵的网格以 0 标注。现在你需要在地图上选择一个矩形区域排兵，这个矩形区域内不能包含不适合排兵的网格。请问可选择的矩形区域最多能包含多少网格？

3.1.2 输入格式

第一行，两个正整数 n, m ，分别表示地图网格的行数与列数。

接下来 n 行，每行 m 个整数 $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ，表示各行中的网格是否适合排兵。

3.1.3 输出格式

一行，一个整数，表示适合排兵的矩形区域包含的最大网格数。

3.1.4 样例

3.1.4.1 输入样例 1

```
1 | 4 3
2 | 0 1 1
3 | 1 0 1
4 | 0 1 1
5 | 1 1 1
```

3.1.4.2 输出样例 1

```
1 | 4
```

3.1.4.3 输入样例 2

```
1 | 3 5
2 | 1 0 1 0 1
3 | 0 1 0 1 0
4 | 0 1 1 1 0
```

3.1.4.4 输出样例 2

```
1 | 3
```

3.1.5 数据范围

对于所有测试点，保证 $1 \leq n, m \leq 12$ ， $0 \leq a_{i,j} \leq 1$ 。

3.1.6 参考程序

```
1 n, m = map(int, input().split())
2 a = []
3 for i in range(n):
4     a.append(list(map(int, input().split())))
5
6 ans = 0
7 for u in range(n):
8     for l in range(m):
9         for d in range(u, n):
10            chk = 1
11            for r in range(l, m):
12                for x in range(u, d + 1):
13                    chk *= a[x][r]
14                    if not chk:
15                        break
16            ans = max(ans, (r - l + 1) * (d - u + 1))
17 print(ans)
```

3.2 编程题 2

- 试题名称：最长连续段
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.2.1 题目描述

对于 k 个整数构成的数组 $[b_1, b_2, \dots, b_k]$ ，如果对 $1 \leq i < k$ 都有 $b_{i+1} = b_i + 1$ ，那么称数组 b 是一个连续段。

给定由 n 个整数构成的数组 $[a_1, a_2, \dots, a_n]$ ，你可以任意重排数组 a 中元素顺序。请问在重排顺序之后， a 所有是连续段的子数组中，最长的子数组长度是多少？

例如，对于数组 $[1, 0, 2, 4]$ ，可以将其重排为 $[4, 0, 1, 2]$ ，有以下 10 个子数组：

$$[4], [0], [1], [2], [4, 0], [0, 1], [1, 2], [4, 0, 1], [0, 1, 2], [4, 0, 1, 2]$$

其中除 $[4, 0], [4, 0, 1], [4, 0, 1, 2]$ 以外的子数组均是连续段，因此是连续段的子数组中，最长子数组长度为 3。

3.2.2 输入格式

第一行，一个正整数 n ，表示数组长度。

第二行， n 个整数 a_1, a_2, \dots, a_n ，表示数组中的整数。

3.2.3 输出格式

一行，一个整数，表示数组 a 重排顺序后，所有是连续段的子数组的最长长度。

3.2.4 样例

3.2.4.1 输入样例 1

```
1 4
2 1 0 2 4
```

3.2.4.2 输出样例 1

```
1 | 3
```

3.2.4.3 输入样例 2

```
1 | 9
2 | 9 9 8 2 4 4 3 5 3
```

3.2.4.4 输出样例 2

```
1 | 4
```

3.2.5 数据范围

对于 40% 的测试点，保证 $1 \leq n \leq 8$ 。

对于所有测试点，保证 $1 \leq n \leq 10^5$ ， $-10^9 \leq a_i \leq 10^9$ 。

3.2.6 参考程序

```
1 n = int(input())
2 a = sorted(list(map(int, input().split())))
3 last, cnt, mx = a[0], 1, 1
4 for i in a:
5     if last == i:
6         continue
7     if last + 1 == i:
8         cnt += 1
9     else:
10        cnt = 1
11        last = i
12        mx = max(mx, cnt)
13 print(mx)
```