



# Python 四级

2025 年 03 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	C	D	C	A	A	D	B	D	A	C	B	C

第 1 题 2025年春节有两件轰动全球的事件，一个是DeepSeek横空出世，另一个是贺岁片《哪吒2》票房惊人，入了全球票房榜。下面关于DeepSeek与《哪吒2》的描述成立的是( )。

- A. 《哪吒2》是一款新型操作系统
- B. DeepSeek是深海钻探软件
- C. 《哪吒2》可以生成新的软件
- D. DeepSeek可以根据《哪吒2》的场景生成剧情脚本

第 2 题 对整型变量N，如果它能够同时被3和5整出，则输出 N是含有至少两个质因数。如果用流程图来描述处理过程，则输出语句应该在何种图形框中 ( )。

- A. 圆形框
- B. 椭圆形框
- C. 平行四边形框
- D. 菱形框

第 3 题 执行下面Python代码后，输出的结果是? ( )

```
1 lst = [{'num': i} for i in range(3)]
2 lst[1]['num'] = 99
3 print(lst)
```

- A. [{'num': 0}, {'num': 1}, {'num': 2}]
- B. [{'num': 0}, {'num': 99}, {'num': 2}]
- C. 所有字典的 'num' 都变为 99
- D. 报错

第 4 题 执行下面Python代码后，输出的结果是? ( )

```
1 keys = ['a', 'b']
2 values = [[1], (2,)]
3 d = {k: v for k, v in zip(keys, values)}
4 d['a'].append(3)
5 print(d['a'])
```

- A. [1]
- B. (2,)
- C. [1, 3]
- D. 报错

第5题 执行下面Python代码后，输出的结果是？（）

```
1 def append_value(value, container=[]):
2     container.append(value)
3     return container
4
5
6 result1 = append_value(1)
7 result2 = append_value(2)
8 result3 = append_value(3, [])
9 print(result1, result2, result3)
```

- A. [1] [2] [3]
- B. [3] [3] [3]
- C. [1] [1, 2] [3]
- D. [1, 2] [1, 2] [3]

第6题 以下哪个函数调用是合法的？（）

```
1 def func(a, b, c=0):
2     print(a + b + c)
```

- A. func(1, b=2, 3)
- B. func(a=1, 2, c=3)
- C. func(1, 2)
- D. func(b=1, a=2, 3)

第7题 执行下面Python代码后，输出的结果是？（）

```
1 def swap(a, b):
2     a, b = b, a
3     return a, b
4
5
6 x, y = 1, 2
7 swap(x, y)
8 print(x, y)
```

- A. 1 2
- B. 2 1
- C. (2, 1)
- D. 报错

第 8 题 执行下面Python代码后，输出的结果是？（ ）

```
1 def outer():
2     n = 0
3
4     def inner():
5         nonlocal n
6         n += 1
7         return n
8
9     return inner
10
11
12 f = outer()
13 print(f(), f(), f())
```

- A. 1 2 3
- B. 1 1 1
- C. 3 3 3
- D. 报错

第 9 题 以下哪个函数调用会返回 [2, 4, 6]？（ ）

```
1 def process(lst, func):
2     return [func(x) for x in lst]
3
4
5 nums = [1, 2, 3]
```

- A. process(nums, lambda x: x % 2)
- B. process(nums, lambda x: x + 1)
- C. process(nums, lambda x: x \*\* 2)
- D. process(nums, lambda x: x \* 2)

第 10 题 执行下面Python代码时，哪条调用会报错？（）

```
1 def func(a, *, b, c):
2     pass
```

- A. func(1, b=2, c=3)
- B. func(1, 2, c=3)
- C. func(a=1, b=2, c=3)
- D. func(1, c=3, b=2)

第 11 题 一只青蛙要跳上 n 级台阶。它每次可以跳1阶、2阶或3阶。下列代码可以计算青蛙有多少种不同的跳跃方式可以到达第 n 阶，其中横线处填写的代码是？（）

```
1 def jump_ways(n):
2     if n <= 3:
3         return [0, 1, 2, 4][n]
4     dp = [0] * (n + 1)
5     dp[1], dp[2], dp[3] = 1, 2, 4
6     for i in range(4, n + 1):
7         dp[i] = _____
8     return dp[n]
```

- A. dp[i - 1] \* 2
- B. dp[i - 3] \* 3
- C. dp[i - 1] + dp[i - 2]
- D. dp[i - 1] + dp[i - 2] + dp[i - 3]

第 12 题 下列代码可将24小时制的"HH:MM"时间字符串转换为从午夜（00:00）开始经过的总分钟数（注：此分钟数为正整数），其中横线处填写的代码是？（）

```
1 time_str = "8:30"
2 h, m = map(_____) # 横线在map的参数部分
3 total_min = h * 60 + m
4 print(total_min) # 应输出510
```

- A. int, time\_str.split(':')
- B. int, time\_str.split(' ')
- C. float, time\_str.split(':')
- D. str, time\_str.split(':')

第 13 题 以下代码实现了哪种排序算法？（）

```
1 def sort(arr):
2     for i in range(len(arr) - 1):
3         min_idx = i
4         for j in range(i + 1, len(arr)):
5             if arr[j] < arr[min_idx]:
6                 min_idx = j
7         arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

- A. 冒泡排序
- B. 插入排序
- C. 选择排序
- D. 快速排序

第 14 题 以下代码的作用是? ( )

```
1 with open(r"data.txt") as f:
2     while True:
3         line = f.readline()
4         if not line:
5             break
6         print(line, end="")
```

- A. 读取全部内容后一次性打印
- B. 逐行读取并打印, 保留原有的换行
- C. 跳过空行打印内容
- D. 倒序打印文件内容

第 15 题 执行下面Python代码后, 输出的结果是? ( )

```
1 def test():
2     try:
3         return "A"
4     finally:
5         return "B"
6
7
8 print(test())
9
```

- A. AB
- B. A
- C. B
- D. 报错

## 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 C++、Python都是高级编程语言，它们每条语句的执行最终都要通过机器指令来完成。（ ）

第 2 题 下列程序用于统计字符串中元音字母（a, e, i, o, u）的数量。

```
1 s = "hello world"
2 vowels = "aeiou"
3 count = sum(1 for c in s if c in vowels)
4 print(count)
```

第 3 题 在函数定义中，\*\*kwargs 必须位于参数列表的最后。（ ）

第 4 题 执行下面Python代码后，调用函数func可以得到一个列表类型的数据。

```
1 def func():
2     return [1, 2],
3
4
5 result = func()
6 print(type(result))
```

第 5 题 Python代码尝试以读取模式（"r"）打开不存在的文件会引发 FileNotFoundError 异常。

第 6 题 下面这段程序的时间复杂度为平方阶  $O(n^2)$ 。

```
1 def func(n):
2     for i in range(n):
3         for j in range(n):
4             print(i, j)
```

第 7 题 对5个不同的数据元素进行直接插入排序，最多需要比较9次。

第 8 题 执行下面Python代码，会触发SyntaxError异常，但不会输出 Error 。

```
1 try:
2     print("Hello"
3 except SyntaxError:
4     print("Error")
```

第 9 题 Python中允许在同一个 with 语句中打开多个文件。

```
1 with open('a.txt', 'r') as f1, open('b.txt', 'w') as f2:
2     data = f1.read()
3     f2.write(data)
```

第 10 题 执行下面Python代码后，会输出 [4, 6] 。

```
1 print(list(map(sum, zip([1, 2], [3, 4]))))
```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：荒地开垦
- 时间限制：3.0 s
- 内存限制：512.0 MB

##### 3.1.1 题面描述

小杨有一大片荒地，可以表示为一个  $n$  行  $m$  列的网格图。

小杨想要开垦这块荒地，但荒地中一些位置存在杂物，对于一块不存在杂物的荒地，该荒地可以开垦当且仅当其上下左右四个方向相邻的格子均不存在杂物。

小杨可以选择至多一个位置，清除该位置的杂物，移除杂物后该位置变为荒地。小杨想知道在清除至多一个位置的杂物的情况下，最多能够开垦多少块荒地。

##### 3.1.2 输入格式

第一行包含两个正整数  $n, m$ ，含义如题面所示。

之后  $n$  行，每行包含一个长度为  $m$  且仅包含字符 `.` 和 `#` 的字符串。如果为 `.`，代表该位置为荒地，如果为 `#`，代表该位置为杂物。

##### 3.1.3 输出格式

输出一个整数，代表在清除至多一个位置的杂物的情况下，最多能够开垦的荒地块数。

##### 3.1.4 样例

##### 3.1.5 输入样例1

```
1 | 3 5
2 | .....
3 | .#..#
4 | .....
```

##### 3.1.6 输出样例1

```
1 | 11
```

##### 3.1.7 样例解释

移除第二行从左数第二块空地的杂物后：

```
1 | .....
2 | ....#
3 | .....
```

第一行从左数前 4 块荒地，第二行从左数前 3 块荒地，第三行从左数前 4 块荒地，均可开垦， $4 + 3 + 4 = 11$ 。

### 3.1.8 数据范围

对于全部数据，保证有  $1 \leq n, m \leq 1000$ 。

### 3.1.9 参考程序

```
1 N = 1005
2 mat = [[''] * N for _ in range(N)]
3 a = [[0] * N for _ in range(N)]
4 d = [(-1, 0), (1, 0), (0, -1), (0, 1)]
5
6
7 n, m = map(int, input().split())
8
9 for i in range(1, n + 1):
10     line = input().strip()
11     for j in range(1, m + 1):
12         mat[i][j] = line[j - 1]
13
14 ans = 0
15
16 for i in range(1, n + 1):
17     for j in range(1, m + 1):
18         num = 0
19         p = -1
20
21         for k in range(4):
22             ni, nj = i + d[k][0], j + d[k][1]
23             if mat[ni][nj] == '#':
24                 num += 1
25                 p = k
26
27         if mat[i][j] == '.' and num == 1:
28             ni, nj = i + d[p][0], j + d[p][1]
29             a[ni][nj] += 1
30         elif mat[i][j] == '.' and num == 0:
31             ans += 1
32         elif mat[i][j] == '#' and num == 0:
33             a[i][j] += 1
34
35 mx = 0
36 for i in range(1, n + 1):
37     for j in range(1, m + 1):
38         mx = max(mx, a[i][j])
39
40 print(ans + mx)
```

## 3.2 编程题 2

- 时间限制：3.0 s
- 内存限制：512.0 MB

### 3.2.10 二阶矩阵

#### 3.2.11 题目描述

小 A 有一个  $n$  行  $m$  列的矩阵  $A$ 。

小 A 认为一个  $2 \times 2$  的矩阵  $D$  是好的，当且仅当  $D_{1,1} \times D_{2,2} = D_{1,2} \times D_{2,1}$ 。其中  $D_{i,j}$  表示矩阵  $D$  的第  $i$  行第  $j$  列的元素。

小 A 想知道  $A$  中有多少个好的子矩阵。

#### 3.2.12 输入格式

第一行，两个正整数  $n, m$ 。

接下来  $n$  行，每行  $m$  个整数  $A_{i,1}, A_{i,2}, \dots, A_{i,m}$ 。

#### 3.2.13 输出格式

一行，一个整数，表示  $A$  中好的子矩阵的数量。

#### 3.2.14 样例

##### 3.2.14.1 输入样例 1

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

##### 3.2.14.2 输出样例 1

```
1 | 2
```

##### 3.2.14.3 样例解释

样例中的好的子矩阵如下：

1	2	1	0
2	4	2	1
0	3	3	0

1	2	1	0
2	4	2	1
0	3	3	0

#### 3.2.15 数据范围

对于所有测试点，保证  $1 \leq n \leq 500$ ， $1 \leq m \leq 500$ ， $-100 \leq A_{i,j} \leq 100$ 。

#### 3.2.16 参考程序

```
1 | n, m = map(int, input().split())
2 | a = [list(map(int, input().split())) for _ in range(n)]
3 | print(sum(1 for i in range(n - 1) for j in range(m - 1) if a[i][j] * a[i + 1][j + 1]
   | == a[i][j + 1] * a[i + 1][j]))
```