



Python 三级

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	A	B	B	B	A	C	D	A	C	C	D	A	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 十进制数13的二进制表示是? ()

- A. 1101
- B. 1001
- C. 1011
- D. 1100

第 4 题 下列代码的输出结果是? ()

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

- A. [1, 2]

- B. [1, 2, 3]
- C. [3, 2, 1]
- D. 报错

第5题 执行 `a = [1, 2]; a.extend([3, 4])` 后, a 的值是? ()

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

第6题 执行 `a, b, c = (10, 20, 30)` 后, b 的值是? ()

- A. 10
- B. 20
- C. 30
- D. 报错

第7题 将列表 `[7, 8, 9]` 转换为元组的正确方式是? ()

- A. `tuple([7, 8, 9])`
- B. `([7, 8, 9]).tuple()`
- C. `(7, 8, 9).tuple()`
- D. `tuple(7, 8, 9)`

第8题 字典的键 (key) 可以是以下哪种数据类型? ()

- A. 列表
- B. 字典
- C. 元组
- D. 集合

第9题 字典推导式 `{k: v for k, v in {'a':1, 'b':2, 'c':3}.items() if v % 2 == 0}` 的结果是? ()

- A. `{'a':1, 'b':2, 'c':3}`
- B. `{}`
- C. `{'a':1, 'c':3}`
- D. `{'b':2}`

第10题 执行 `s = "hello"; print(s.find("x"))` 的结果是? ()

- A. -1

- B. 0
- C. 报错
- D. None

第 11 题 字符串 `s = "Python"`，执行 `s[::-2]` 的结果是？（）

- A. "Pyt"
- B. "yhn"
- C. "Pto"
- D. "yh"

第 12 题 创建空集合的正确方式是？（）

- A. `a = []`
- B. `a = {}`
- C. `a = set()`
- D. B和C均正确

第 13 题 执行 `s = {1, 2}; s.update([3, 4])` 后，`s` 的值是？（）

- A. 报错
- B. `{3, 4}`
- C. `{1, 2, [3, 4]}`
- D. `{1, 2, 3, 4}`

第 14 题 需要将文本中的敏感词替换为保留首字母并用 * 号覆盖剩余字符。例如：“密码”（2字）替换为“密*”，“身份证”（3字）替换为“身**”。请填写横线处的代码。

```
1 words = ["密码", "账号", "身份证"]
2 text = "请勿泄露您的密码和身份证号码"
3 for word in words:
4     if word in text:
5         text = text.replace(word, _____)
6 print(text) # 期望输出"请勿泄露您的密*和身**号码"
```

- A. `word[0] + '*' * (len(word) - 1)`
- B. `word[0] + '**'`
- C. `'*' * len(word)`
- D. `word[0] + '*' * len(word)`

第 15 题 以下代码用于生成前 `n` 项斐波那契数列。请填写横线处的代码。

```

1 n = 5
2 a, b = 0, 1
3 fib = []
4 for _ in range(n):
5     fib.append(a)
6     a, b = b, _____
7 print(fib) # 期望输出 [0, 1, 1, 2, 3]

```

- A. a
- B. b
- C. a + b
- D. a * b

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 表达式 `hex(int('17', 8))` 的返回值为 `0x11`。

第 3 题 在 Python 中，`in` 关键词用于检查某个元素是否存在于序列或集合中，以下表达式都是合法的。

```

1 "a" in "gesp"
2 "a" in {"g", "e", "s", "p"}
3 "a" in ["g", "e", "s", "p"]
4 "a" in ("g", "e", "s", "p")
5 "a" in {1: "g", 2: "e", 3: "s", 4: "p"}

```

第 4 题 表达式 `y = 0x0f; y << 4` 的结果是 `0xff`。

第 5 题 执行下面 Python 代码后，输出的结果为 6。

```

1 data = {'ids': [1, 2], 'name': 'test'}
2 data['ids'].append(['g', 'e', 's', 'p'])
3 print(len(data['ids']))

```

第 6 题 执行下面 Python 代码后，输出的结果可能是 `{'a', 1, '123'}`。

```

1 lst = ['a', 'a', 1, 1]
2 s = set(lst)
3 s.update('123')
4 print(s)

```

第 7 题 自幂数是指一个 `n` 位数，它的每个位上的数字的 `n` 次幂之和等于它本身。（例如：当 `n` 为 3 时，有 $1^3 + 5^3 + 3^3 = 153$ ，153 即是 `n` 为 3 时的一个自幂数），若找出 1000 以内的所有自幂数。可以使用枚举的方法来求解这个问题。

第 8 题 下列程序用于删除列表 `nums` 中的重复项。

```
1 nums = [1, 2, 2, 3, 4, 4]
2 unique = list(set(nums))
3 print(unique)
```

第9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 == vowels)
4 print(count)
```

第10题 下列程序用于去除列表中的负整数元素。

```
1 raw = [1, -2, 3, 5, -7, 0]
2 clean = [x for x in raw if x >= 0]
3 print(clean)
```

3 编程题 (每题 25 分, 共 50 分)

3.1 编程题 1

- 时间限制: 1.0 s
- 内存限制: 512.0 MB

3.1.1 2025

3.1.2 题目描述

小 A 有一个整数 x , 他想找到最小的正整数 y 使得下式成立:

$$(x \text{ and } y) + (x \text{ or } y) = 2025$$

其中 `and` 表示二进制按位与运算, `or` 表示二进制按位或运算。如果不存在满足条件的 y , 则输出 -1 。

3.1.3 输入格式

一行, 一个整数 x 。

3.1.4 输出格式

一行, 一个整数, 若满足条件的 y 存在则输出 y , 否则输出 -1 。

3.1.5 样例

3.1.5.1 输入样例 1

```
1 | 1025
```

3.1.5.2 输出样例 1

```
1 | 1000
```

3.1.6 数据范围

对于所有测试点，保证 $0 \leq x < 2025$ 。

3.1.7 提示

$$(x \text{ and } y) + (x \text{ or } y) = 2025$$

其中：

- and 表示按位与运算，运算符为 &。
- or 表示按位或运算，运算符为 |。

3.1.8 参考程序

```
1 x = int(input())
2 for y in range(1,2026):
3     if (x&y)+(x|y) == 2025:
4         print(y)
5         exit(0)
6 print(-1)
```

3.2 编程题 2

- 时间限制：1.0 s
- 内存限制：512.0 MB

3.2.9 词频统计

3.2.10 题目描述

在文本处理中，统计单词出现的频率是一个常见的任务。现在，给定 n 个单词，你需要找出其中出现次数最多的单词。在本题中，忽略单词中字母的大小写（即 Apple、apple、APPLE、aPPlE 等均视为同一个单词）。

请你编写一个程序，输入 n 个单词，输出其中出现次数最多的单词。

3.2.11 输入格式

第一行，一个整数 n ，表示单词的个数；

接下来 n 行，每行包含一个单词，单词由大小写英文字母组成。

输入保证，出现次数最多的单词只会一个。

3.2.12 输出格式

输出一行，包含出现次数最多的单词（输出单词为小写形式）。

3.2.13 样例

3.2.13.3 输入样例 1

```
1 6
2 Apple
3 banana
4 apple
5 Orange
```

```
7 | apple
```

3.2.13.4 输出样例 1

```
1 | apple
```

3.2.14 数据范围

对于所有测试点， $1 \leq n \leq 100$ ，每个单词的长度不超过 30，且仅由大小写英文字母组成。

3.2.15 参考程序

```
1 | n = int(input())
2 | mx, cnt, ans = -1, {}, ""
3 |
4 | for i in range(n):
5 |     s = input().lower()
6 |     if s not in cnt:
7 |         cnt[s] = 0
8 |     cnt[s] += 1
9 |     if cnt[s] > mx:
10 |         mx, ans = cnt[s], s
11 |
12 | print(ans)
```