



Python 三级

2026 年 06 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	C	B	A	A	B	D	C	B	B	B	A	C	C	A	A

第 1 题 学校组织同学们到未来农场参观，小明听讲解员介绍。在智能温室中，湿度传感器可以连续检测土壤的湿度，并将检测到的湿度数据实时发送给中央控制器。中央控制器根据这些数据判断是否开启灌溉系统。请问，这里的湿度传感器所发挥的作用，类似于计算机系统中的哪一类组件？（ ）

- A. 处理器
- B. 存储器
- C. 输入设备
- D. 输出设备

第 2 题 小明同学去郊外农场参观回来后，在自己的电脑中安装了一个 3D 农场仿真模拟系统。因为他今年买的这台二手电脑有点老旧，3D 农场仿真模拟系统运行时经常弹出“系统内存不足”的警告，导致系统卡顿严重。他计划通过增加物理内存来解决问题。请问他应该购买以下哪种硬件组件？（ ）

- A. 机械硬盘
- B. 内存条
- C. 图形显卡
- D. 移动硬盘

第 3 题 在 Python 中，`-1` 的 16 位补码用十六进制表示是？（ ）

- A. `0xFFFF`
- B. `0x8001`
- C. `0x0001`
- D. `0xFFFE`

第 4 题 执行 `print(bin(0o15))` 的输出结果是？（ ）

- A. `0b1101`
- B. `0b1110`
- C. `0b1100`
- D. `0b1111`

第 5 题 下列哪个表达式可以将整数 `x` 的二进制表示中从右往左数第 5 位（最低位为第 0 位）强制设置为 1，而其他位保持不变？（ ）

- A. `x & (~(1 << 5))`
- B. `x | (1 << 5)`
- C. `x ^ (1 << 5)`
- D. `x >> 5`

第6题 下面 Python 代码执行后，输出的结果是？（ ）

```
1 lst = [1, 2, 3, 4, 5]
2 print(lst.pop(-2))
3 print(lst)
```

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

第7题 `t = (2, 4, 6, 2)`，执行 `t.count(2)` 的结果是？（ ）

- A. 0
- B. 1
- C. 2
- D. 4

第8题 下面 Python 代码执行后，输出的结果是？（ ）

```
1 dict_a = {'a': 1, 'b': 2, 'c': 3}
2 dict_b = {'b': 4, 'd': 5}
3 dict_a.update(dict_b)
4 print(dict_a['b'])
```

- A. 2
- B. 4
- C. 5
- D. 报错

第9题 执行 `s = "a,b,c,d"; print(s.split(',', 2))` 的结果是？（ ）

- A. ['a', 'b', 'c', 'd']
- B. ['a', 'b', 'c,d']
- C. ['a', 'b,c,d']
- D. ['a,b', 'c,d']

第10题 已知 `set1 = {1, 2}`，`set2 = {3}`。现要求在不修改 `set1` 的前提下，获得一个包含 `set1` 和 `set2` 所有元素的新集合，以下哪个操作是正确的？（ ）

- A. `set1 + set2`
- B. `set1.union(set2)`
- C. `set1.append(set2)`
- D. `set1.update(set2)`

第 11 题 从字典 `d = {'a': 1, 'b': 2}` 中，如何得到一个只包含键的列表？（ ）

- A. `list(d)`
- B. `d.keys()`
- C. `d.to_list()`
- D. `list(d.items())`

第 12 题 用 1 元、2 元、5 元硬币凑出 10 元，且硬币总数不超过 5 枚。以下程序计算符合要求的组合个数，横线处应填写的代码是？（ ）

```
1 count = 0
2 for one in range(0, 11):
3     for two in range(0, 6):
4         for five in range(0, 3):
5             total_money = one + two * 2 + five * 5
6             total_coins = one + two + five
7             if _____:
8                 count += 1
9 print(count)
```

- A. `total_money == 10`
- B. `total_coins <= 5`
- C. `total_money == 10 and total_coins <= 5`
- D. `total_money == 10 and total_coins >= 5`

第 13 题 下列程序能够将列表中的偶数筛选出来，并组成新列表。横线处填写的代码是？（ ）

```
1 nums = [1, 2, 3, 4, 5, 6]
2 result = [x for x in nums if _____]
3 print(result) # 期望输出 [2, 4, 6]
```

- A. `x * 2`
- B. `x % 2 != 0`
- C. `x % 2 == 0`
- D. `x / 2 == 0`

第 14 题 某电商平台有 M 种商品类别，平台记录了 N 条销售记录，每条记录包含商品类别编号和销售数量。需要找出每个类别中单次销售数量最多的那一笔（即该类别的最大销售数量），并输出这些最大数量中的最小值。以下程序试图实现此功能，横线处填写的代码是？（ ）

```
1 M, N = map(int, input().split())
2 max_sale = [None] * (M + 1) # 用 None 初始化, 表示暂无记录
3 for _ in range(N):
4     cate, amount = map(int, input().split())
5     _____
6     max_sale[cate] = amount
7
8 valid_max = [v for v in max_sale[1:] if v is not None]
9 result = min(valid_max)
10 print(result)
```

- A. `if max_sale[cate] is None or amount > max_sale[cate]:`
- B. `if amount > max_sale[cate]:`

- C. `if max_sale[cate] is None:`
- D. `while amount > max_sale[cate]:`

第 15 题 以下程序实现了一种简单的“翻转”加密：将大写字母表中的每个字母替换为与其对称的字母（A ↔ Z、B ↔ Y、C ↔ X、……）。横线处填写的代码是？（ ）

```

1 text = "HELLO"
2 result = []
3 for ch in text:
4     if 'A' <= ch <= 'Z':
5         -----
6         result.append(new_ch)
7     else:
8         result.append(ch)
9 encrypted = ''.join(result)
10 print(encrypted)

```

- A. `new_ch = chr(ord('Z') - (ord(ch) - ord('A')))`
- B. `new_ch = chr(ord('Z') - ord(ch) - ord('A'))`
- C. `new_ch = chr(ord('Z') + (ord(ch) - ord('A')))`
- D. `new_ch = chr(ord('Z') - ord(ch))`

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 又到期末考试周，小明发现这次许多闭卷考试不仅禁止携带手机、平板电脑，还有最近比较时髦的各类 AI 眼镜（也有叫智能眼镜）也同样不允许带入考场。这些 AI 眼镜应该也是内置了操作系统并可能支持 Wi-Fi 或蓝牙连接。

第 2 题 `bin()`、`oct()`、`hex()` 这三个函数的返回值类型是整数（`int`）。

第 3 题 表达式 `(x & (x-1))` 可以用来判断正整数 `x` 是否为 2 的幂次方。

第 4 题 切片操作 `mylist[2:5:-1]` 会返回索引 2 到索引 4（包含）的元素，但顺序是反向的。

第 5 题 创建只包含一个元素的元组时，必须在元素后加一个逗号，例如 `(1,)`。

第 6 题 执行 `d1 = {x: x for x in [1, 1, 2]}` 后，`d1` 为 `{1: 1, 2: 2}`。

第 7 题 使用 `in` 运算符检查子字符串是否存在时，如果字符串为空，`"" in "abc"` 将返回 `False`。

第 8 题 执行下面 Python 代码后，输出的结果为 2。

```

1 s = {10, 20, 30}
2 s.remove(20)
3 s.discard(40)
4 print(len(s))

```

第 9 题 执行下面 Python 代码后，输出的结果为 7。

```
1 my_str = "hello world"
2 my_set = set(my_str)
3 result = len(''.join(my_set))
4 print(len)
```

第 10 题 下列 Python 代码能够正确统计出所有能被 3 整除且至少包含一个数字 5 的四位数的个数。

```
1 count = 0
2 for num in range(1000, 10000):
3     if num % 3 == 0:
4         digits = [int(d) for d in str(num)]
5         if 5 in digits:
6             count += 1
7 print(count)
```

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称：加密
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.1.1 题目描述

小杨同学有一串数字，想把它们变成另一串数字，这个过程叫做**加密**。

他有一本**密码本**，密码本告诉你：

每个数字应该变成哪个数字。

数字一共有 10 个：

0、1、2、3、4、5、6、7、8、9

密码本会依次告诉你：

0 要变成什么

1 要变成什么

2 要变成什么

.....

9 要变成什么

请你按照密码本，把原来的每个数字都换成新的数字，然后输出。

3.1.2 输入格式

输入共有 3 行。

第一行：一个整数，表示有多少个数字需要加密；

第二行：这些需要加密的数字；

第三行：密码本，一共 10 个数字。

这 10 个数字的意思是：

第 1 个数字：表示 0 加密后变成什么；

第 2 个数字：表示 1 加密后变成什么；

第 3 个数字：表示 2 加密后变成什么；

.....

第 10 个数字：表示 9 加密后变成什么。

3.1.3 输出格式

输出加密后的数字。

也就是：把输入第二行里的每个数字，都按照输入第三行的密码本换掉后输出。

3.1.4 样例

3.1.5 输入样例 1

```
1 | 7
2 | 0 2 0 3 4 1 9
3 | 9 0 1 2 3 4 5 6 7 8
```

3.1.6 输出样例 1

```
1 | 9 1 9 2 3 0 8
```

3.1.7 样例解释

第二行要加密的数字是：

0 2 0 3 4 1 9

第三行密码本是：

9 0 1 2 3 4 5 6 7 8

它的意思是：

0 变成 9

1 变成 0

2 变成 1

3 变成 2

4 变成 3

5 变成 4

6 变成 5

7 变成 6

8 变成 7

9 变成 8

所以：

0 变成 9

2 变成 1

0 变成 9

3 变成 2

4 变成 3

1 变成 0

9 变成 8

最后得到：

```
9 1 9 2 3 0 8
```

3.1.8 数据范围

需要加密的数字个数不超过 20000 个，且均为 0 到 9；密码本中的数字不重复，且均为 0 到 9。

3.1.9 参考程序

```
1 # 需要加密的数字
2 n = int(input())
3 # 需要被加密的文本
4 txt = input().split()
5 # 密码本
6 keys = list(input().split())
7 for x in txt:
8     # keys[x]: 将原始文本映射成加密后的
9     print(keys[int(x)], end = ' ')
```

3.2 编程题 2

- 试题名称：字符转换
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.2.1 题目描述

小杨同学有一串字符，里面可能有：

- 大写字母，比如 A、B、C
- 小写字母，比如 a、b、c
- 数字，比如 0、1、2

现在小杨同学想把这串字符变一变，规则如下：

1. 如果是大写字母，就变成对应的小写字母；
2. 如果是小写字母，就变成对应的大写字母；
3. 如果是数字，就变成 *。

请你按照这个规则，帮小杨把整串字符转换好。

3.2.2 输入格式

输入一共有 2 行。

第一行：一个整数，表示这串字符一共有多少个字符。

第二行：一串连续的字符，中间没有空格。

3.2.3 输出格式

输出转换后的字符。

注意：

输出时字符之间不要加空格。

3.2.4 样例

3.2.5 输入样例 1

```
1 | 5
2 | aBc98
```

3.2.6 输出样例 1

```
1 | AbC**
```

3.2.7 样例解释

原来的字符是：

```
1 | aBc98
```

从左到右一个一个看：

1. a 是小写字母，所以变成 A ；
2. B 是大写字母，所以变成 b ；
3. c 是小写字母，所以变成 C ；
4. 9 是数字，所以变成 * ；
5. 8 是数字，所以变成 * 。

所以最后输出：

```
1 | AbC**
```

3.2.8 数据范围

字符个数不会超过 1000 个。

每个字符只会是大写字母、小写字母或数字。

3.2.9 参考程序

```
1 # 文本长度n
2 n = int(input())
3 # 需要转换的字符串
4 s = input().strip()
5 # 最终的答案
6 ans = ''
7 # 遍历字符串中的字符
8 for c in s:
9     # 如果是小写就变成大写
10    if 'a' <= c <= 'z':
11        ans += c.upper()
12    # 如果c是大写就变成小写
13    elif 'A' <= c <= 'Z':
14        ans += c.lower()
15    # 如果是数字就变成*
16    else:
17        ans += '*'
18 print(ans)
```