



Python 三级

2025 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	A	B	A	C	D	A	D	B	B	C	C	B	A	A

第 1 题 2025年4月19日在北京举行了一场颇为瞩目的人形机器人半程马拉松赛。比赛期间，跑动着的机器人会利用身上安装多个传感器所反馈的数据来调整姿态、保持平衡等，那么这类传感器类似于计算机的()。

- A. 处理器
- B. 存储器
- C. 输入设备
- D. 输出设备

第 2 题 小杨购置的计算机使用一年后觉得内存不够用了，想购置一个容量更大的内存条，这时需要的内存条是()。

- A. RAM
- B. ROM
- C. CACHE
- D. EPROM

第 3 题 下列哪个Python表达式能正确将十六进制数"2A"转换为十进制? ()

- A. `int('2A', 10)`
- B. `int('2A', 16)`
- C. `bin(0x2A)`
- D. `hex(42)`

第 4 题 执行下面Python代码后，输出的结果是? ()

```
1 | fruits = ['apple', 'banana', 'cherry']  
2 | fruits.append('orange')  
3 | fruits.insert(1, 'grape')  
4 | print(fruits)
```

- A. `['apple', 'grape', 'banana', 'cherry', 'orange']`
- B. `['apple', 'banana', 'grape', 'cherry', 'orange']`

- C. ['apple', 'grape', 'banana', 'cherry']
- D. ['apple', 'grape', 'orange', 'banana', 'cherry']

第5题 执行下面Python代码后，输出的结果是？（）

```
1 numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
2 result = numbers[2:7:2] + numbers[-3::-3]
3 print(result)
```

- A. [2, 4, 6, 7, 4]
- B. [2, 4, 6, 6, 3, 0]
- C. [2, 4, 6, 7, 4, 1]
- D. [2, 4, 6, 7, 4, 1, 0]

第6题 关于Python元组，以下说法错误的是？（）

- A. 元组是不可变的数据类型
- B. 元组可以作为字典的键
- C. 元组支持切片操作
- D. 元组中的元素可以被重新赋值

第7题 执行下面Python代码后，输出的结果是？（）

```
1 print((1, 2, 3) < (1, 2, 4))
```

- A. True
- B. False
- C. 报错
- D. None

第8题 关于字典的 .get() 方法，以下说法错误的是？（）

- A. 当键不存在时，.get() 方法可以返回 None
- B. .get() 方法可以设置默认返回值
- C. .get() 方法比直接使用 [] 访问更安全
- D. .get() 方法会修改原字典

第9题 以下哪个字典推导式能正确将列表 ['Python', 'Java', 'C++'] 转换为 {'Python':6, 'Java':4, 'C++':3} (字符串作为键，长度作为值)？（）

- A. {len(x): x for x in ['Python', 'Java', 'C++']}
- B. {x: len(x) for x in ['Python', 'Java', 'C++']}
- C. {x: x.length() for x in ['Python', 'Java', 'C++']}

- D. {x: size(x) for x in ['Python', 'Java', 'C++']}

第10题 执行下面Python代码后，输出的结果是？（）

```
1 s = " Hello, World! "  
2 result = s.strip().lower().replace('world', 'Python').capitalize()  
3 print(result)
```

- A. Hello, python!
 B. Hello, Python!
 C. hello, python!
 D. hello, Python!

第11题 执行下面Python代码后，输出的结果是？（）

```
1 s = "one,two,three"  
2 parts = s.split(',')  
3 new_s = '-'.join([p[:2] for p in parts])  
4 print(new_s)
```

- A. o-t-t
 B. one-two-three
 C. on-tw-th
 D. one,two,three

第12题 以下哪个操作会修改集合 `s = {1, 2, 3}` 的内容？（）

- A. `s.union({4, 5})`
 B. `s.difference({1})`
 C. `s.intersection_update({2, 3})`
 D. `s.issubset({1, 2, 3, 4})`

第13题 执行下面Python代码后，输出的结果是？（）

```
1 set1 = {1, 2, 3}  
2 set2 = {3, 4, 5}  
3 print(set1 ^ set2)
```

- A. {1, 2, 3, 4, 5}
 B. {1, 2, 4, 5}
 C. {3}
 D. set()

第14题 下列哪个选项可以将字符串 "hello" 转换为列表 ['h', 'e', 'l', 'l', 'o']？（）

- A. `list("hello")`

- B. "hello".split()
- C. "hello".to_list()
- D. ["hello"]

第 15 题 凯撒加密是一种经典的加密方法，它通过将字母表中的每个字母按照固定的位移量进行替换来实现加密。例如，当位移量为1时：

- a→b, b→c, ..., y→z
- z→a（循环回到字母表开头）
- 其他字符（如大写字母、数字、符号等）保持不变

现在需要实现一个凯撒加密程序，具体要求如下：

- 只对小写字母进行加密
- 位移量为1
- 非小写字母字符保持原样不变
- 加密后的字符需要保持原始的大小写格式

请补全下面代码中的空白处，使其能够正确实现上述加密功能：

```

1 text = input("请输入要加密的字符串：")
2 encrypted = []
3 for char in text:
4     if char.islower():
5         code = ord(char) # 获取字符的ASCII码
6         new_code = (_____) % 26 + ord('a') # 填空处
7         encrypted.append(chr(new_code))
8     else:
9         encrypted.append(char)
10 print("加密结果：", ''.join(encrypted))

```

- A. code - ord('a') + 1
- B. code + 1
- C. ord(char) + 1
- D. (code - ord('a')) * 2

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 现在，人们参加各种闭卷考试时通常都不允许将智能手机、平板电脑等带入考场，因为智能手表通常都有嵌入操作系统及通信等功能，所以也不允许携带入考场。（ ）

第 2 题 执行下面Python代码后，输出的结果是 True。

```
1 | x = 15
2 | y = 7
3 | print((x ^ y) < (x & y))
```

第3题 表达式 `~0b0011 << 2` 的结果是 `-0b1100` (即十进制 `-12`)。

第4题 若用枚举算法解决"密码锁破解"问题, 则十进制3位数字密码最多需要尝试1000次就一定能打开。

第5题 执行下面Python代码后, 会将列表 `a` 的前两个元素替换为 `[10, 20]`。

```
1 | a = [1, 2, 3, 4, 5]
2 | a[:2] = [10, 20]
3 | print(a)
```

第6题 使用 `update()` 方法合并字典时, 如果键重复, 新值会覆盖旧值。

第7题 执行下列Python代码, 输出的结果是2。

```
1 | t = (1, 2, 3) * 2
2 | print(len(t))
```

第8题 Python代码 `set1 = {1, 2, 3}; set1.remove(4)` 能正常执行, 且 `set1` 保持不变。

第9题 在Python中, 字符串是不可变对象, 因此执行 `s = "hello"; s[0] = "H"` 会导致程序报错。

第10题 在Python中使用 `list({"a": 1, "b": 2})` 可以将字典 `{"a": 1, "b": 2}` 转换为 `[("a", 1), ("b", 2)]`。

3 编程题 (每题 25 分, 共 50 分)

3.1 编程题 1

- 试题名称: 奇偶校验
- 时间限制: 3.0 s
- 内存限制: 512.0 MB

3.1.1 题目描述

数据在传输过程中可能出错, 因此接收方收到数据后通常会校验传输的数据是否正确, 奇偶校验是经典的校验方式之一。

给定 n 个非负整数 c_1, c_2, \dots, c_n 代表所传输的数据, 它们的校验码取决于这些整数在二进制下 1 的数量之和的奇偶性。如果这些整数在二进制下共有奇数个 1, 那么校验码为 1; 否则校验码为 0。你能求出这些整数的校验码吗?

3.1.2 输入格式

第一行, 一个正整数 n , 表示所传输的数据量。

第二行, n 个非负整数 c_1, c_2, \dots, c_n , 表示所传输的数据。

3.1.3 输出格式

输出一行，两个整数，以一个空格分隔：

第一个整数表示 c_1, c_2, \dots, c_n 在二进制下 1 的总数量；

第二个整数表示校验码（0或1）。

3.1.4 样例

3.1.4.1 输入样例 1

```
1 | 4
2 | 71 69 83 80
```

3.1.4.2 输出样例 1

```
1 | 13 1
```

3.1.4.3 输入样例 2

```
1 | 6
2 | 1 2 4 8 16 32
```

3.1.4.4 输出样例 2

```
1 | 6 0
```

3.1.5 数据范围

对于所有测试点，保证 $1 \leq n \leq 100$ ， $0 \leq c_i \leq 255$ 。

3.1.6 参考程序

```
1 n = int(input()) # 随后将输入的数据个数
2 nums = input().split() # 输入的数据行
3 nums = list(map(int,nums)) # 将每个数都转换为整数，最后转换为list
4 one_count = 0
5 for i in nums:
6     string_bin = bin(i)[2:] # 注意转换为二进制后，前导有0b，如bin(10)的结果为0b1010
7     one_count += string_bin.count("1") # 统计1的数量，并累加到one_count变量
8 # 求出校验码
9 if one_count % 2 == 1:
10     check_number = 1 # 奇数时
11 else:
12     check_number = 0 # 偶数时
13 print(one_count,check_number) # 按要求输出
```

3.2 编程题 2

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

3.2.1 题目描述

有 n 位小朋友排成一队等待老师分糖果。第 i 位小朋友想要至少 a_i 颗糖果，并且分给他的糖果数量必须比分给前一位小朋友的糖果数量更多，不然他就会不开心。

老师想知道至少需要准备多少颗糖果才能让所有小朋友都开心。你能帮帮老师吗？

3.2.2 输入格式

第一行，一个正整数 n ，表示小朋友的人数。

第二行， n 个正整数 a_1, a_2, \dots, a_n ，依次表示每位小朋友至少需要的糖果数量。

3.2.3 输出格式

输出一行，一个整数，表示最少需要准备的糖果数量。

3.2.4 样例

3.2.4.1 输入样例 1

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

3.2.4.2 输出样例 1

```
1 | 16
```

3.2.4.3 输入样例 2

```
1 | 15
2 | 314 15926 53589793 238462643 383279502 8 8 4 1 9 7 1 6 9 3
```

3.2.4.4 输出样例 2

```
1 | 4508143253
```

3.2.5 数据范围

对于所有测试点，保证 $1 \leq n \leq 1000$ ， $1 \leq a_i \leq 10^9$ 。

3.2.6 参考程序

```
1 | n = int(input()) # 随后将输入的数据个数
2 | sugar_nums = list(map(int, input().split())) # 输入的数据行，并先转换为整数后转换为list
3 | student_count = len(sugar_nums) # 小朋友数量
4 | for i in range(1, student_count):
5 |     # 如果小于等于前一位小朋友
6 |     if sugar_nums[i] <= sugar_nums[i - 1]:
7 |         sugar_nums[i] = sugar_nums[i - 1] + 1
8 | print(sum(sugar_nums)) # 输出求和值，如果不知道sum()函数，则循环累加求和
```