



Python 三级

2026 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	C	B	D	B	A	C	C	B	A	D	B	B	A	D	C

第 1 题 第二届世界人形机器人运动会开幕式上，机器人方阵通过控制程序来完成整场表演。表演前，工程师们使用集成开发环境（IDE）编写并调试方阵的控制程序。下列关于 IDE 功能的描述，错误的是（ ）。

- ☐ A. 可以提供代码自动补全，提高编程效率
- ☐ B. 可以提供辅助信息，帮助排查程序逻辑错误
- ☐ C. 可以直接控制机器人电机转动
- ☐ D. 支持语法高亮，使代码更易阅读

第 2 题 计算机 CPU 中的高速缓冲存储器（Cache）主要为提升系统性能而设计。下列关于 Cache 的说法中正确的是（ ）。

- ☐ A. Cache 的容量比主存大，速度比主存快
- ☐ B. Cache 通常用于暂存 CPU 频繁访问的数据
- ☐ C. Cache 中的数据断电后不会丢失，属于非易失性存储器
- ☐ D. Cache 的访问速度比 CPU 寄存器还要快

第 3 题 8 位二进制原码能表示的最小和最大整数分别是？（ ）

- ☐ A. -128 128
- ☐ B. -128 127
- ☐ C. -127 128
- ☐ D. -127 127

第 4 题 在 Python 中，表达式 `-0b1111` 的十进制值是？（ ）

- ☐ A. 15
- ☐ B. -15
- ☐ C. -1
- ☐ D. -7

第 5 题 在 Python 中，表达式 `~5+1` 的结果是？（ ）

- ☐ A. -5
- ☐ B. 6

☐ C. -6

☐ D. 4

第6题 执行下面 Python 代码后，输出的结果是？（ ）

```
1 my_list = ['a', 'b', 'c', 'd', 'e', 'f']
2 result = my_list[::2] + my_list[1::2]
3 print(result)
```

☐ A. ['a', 'c', 'e']

☐ B. ['b', 'd', 'f']

☐ C. ['a', 'c', 'e', 'b', 'd', 'f']

☐ D. ['a', 'b', 'c', 'd', 'e', 'f']

第7题 执行下面 Python 代码后，输出的结果是？（ ）

```
1 numbers = [5, 2, 8, 1, 9]
2 i = 0
3 while i < len(numbers):
4     if numbers[i] < 5:
5         numbers.pop(i)
6     else:
7         i += 1
8 print(len(numbers))
```

☐ A. 1

☐ B. 2

☐ C. 3

☐ D. 5

第8题 执行下面 Python 代码后，输出的结果是？（ ）

```
1 my_tuple = (10, 20, 30, 40, 50)
2 a, *b, c = my_tuple
3 print(b)
```

☐ A. 20

☐ B. [20, 30, 40]

☐ C. (20, 30, 40)

☐ D. 上述代码执行报错

第9题 执行下面 Python 代码后，输出的结果是？（ ）

```
1 dict1 = {'a': 1, 'b': 2}
2 dict2 = dict1
3 dict2['c'] = 3
4 print(dict1 is dict2, dict1 == dict2)
```

☐ A. True True

☐ B. True False

☐ C. False True

☐ D. False False

第 10 题 执行下面 Python 代码，输入 1a2b3c4，输出的结果是？（ ）

```
1 s = input("请输入字符串: ")
2 count = 0
3 for i in range(len(s) - 1):
4     if s[i].isdigit() and s[i + 1].isalpha():
5         count += 1
6 print(count)
```

- ☐ A. 0
- ☐ B. 1
- ☐ C. 2
- ☐ D. 3

第 11 题 在 Python 中，表达式 {x % 3 for x in range(6)} 的结果是？（ ）

- ☐ A. {0, 1, 2, 3}
- ☐ B. {0, 1, 2}
- ☐ C. {0, 1}
- ☐ D. {3}

第 12 题 执行下面 Python 代码后，输出的结果是？（ ）

```
1 my_tuple = (1, 2, 3)
2 my_list = list(my_tuple)
3 my_list.append(4)
4 new_tuple = tuple(my_list)
5 print(new_tuple)
```

- ☐ A. (1, 2, 3)
- ☐ B. (1, 2, 3, 4)
- ☐ C. [1, 2, 3, 4]
- ☐ D. 代码会报错，因为元组不可修改

第 13 题 笔记本 5 元/本，笔 3 元/支，橡皮 1 元/个，用 100 元买 50 件文具。以下程序枚举所有购买方案，横线处应填写的代码是？（ ）

```
1 for notebook in range(0, 21):
2     for pen in range(0, 34):
3         for eraser in range(0, 51):
4             if _____:
5                 print(notebook, pen, eraser)
```

- ☐ A. notebook + pen + eraser == 50 and notebook*5 + pen*3 + eraser*1 == 100
- ☐ B. notebook + pen + eraser == 50
- ☐ C. notebook*5 + pen*3 + eraser*1 == 100
- ☐ D. notebook*5 + pen*3 + eraser*1 <= 100

第 14 题 以下 Python 代码的功能是？（ ）

```

1 s = input("请输入字符串: ")
2 words = s.split()
3 longest_word = ""
4 for word in words:
5     if len(word) > len(longest_word):
6         longest_word = word
7 print(longest_word)

```

- ☐ A. 找出字符串中出现次数最多的单词
- ☐ B. 找出字符串中长度最大的单词，若有多个则全部输出
- ☐ C. 找出字符串中最后一个出现的长度最大的单词
- ☐ D. 找出字符串中第一个出现的长度最大的单词

第 15 题 学校举行编程比赛，共有 M 个班级参加，总计 N 名学生提交了作品。每份作品有一个班级编号（1 到 M ）和得分。现需要统计每个班级的最高得分，并计算所有班级最高得分的总和。分数为非负数。以下程序试图实现此功能，横线处应填写的代码是？（ ）

```

1 M, N = map(int, input().split())
2 max_score = [0] * (M + 1)
3 for _ in range(N):
4     cls, score = map(int, input().split())
5     -----①-----
6     max_score[cls] = score
7 -----②-----
8 print(total)

```

- ☐ A. ① `if score > max_score[cls]:` , ② `total = sum(max_score[0:M])`
- ☐ B. ① `if score < max_score[cls]:` , ② `total = sum(max_score[1:])`
- ☐ C. ① `if score > max_score[cls]:` , ② `total = sum(max_score[1:])`
- ☐ D. ① `if score >= max_score[cls]:` , ② `total = sum(scores)`

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 Python 程序执行 `input()` 函数时通常是使用键盘等输入设备来输入某个变量的值。

第 2 题 执行下面 Python 代码后，输出的结果是 C3。

```

1 s = "10110011"
2 num = int(s, 2)
3 hex_str = hex(num)[2:].upper()
4 print(hex_str)

```

第 3 题 对于任意整数 x ， $x \wedge (\sim x)$ 的结果恒为 -1。

第 4 题 下列程序用于输出 1 至 20 中，2 和 3 的共同倍数。

```

1 a = [i + 1 for i in range(20)]
2 for i in range(20):
3     if a[i] % 2 or a[i] % 3:
4         a[i] = 0
5 for i in range(20):
6     if a[i]:
7         print(a[i], end=" ")

```

第5题 执行下面两段 Python 代码后，输出结果是相同的。

```

1 b = [1, 2, 3]
2 c = b
3 b = b[:-1]
4 print(c)

```

```

1 b = [1, 2, 3]
2 c = b
3 b[:] = b[:-1]
4 print(c)

```

第6题 执行下面 Python 代码后，输出的结果是 10。

```

1 a = len(x for x in range(10))
2 print(a)

```

第7题 字典的键可以是列表，例如 `{[1,2]: 'a'}` 在 Python 中不会引发错误。

第8题 执行下面 Python 代码后，输出的结果是 `['a', 'b']`。

```

1 s = "a..b".split('.')
2 print(s)

```

第9题 执行下面 Python 代码后，输出的结果是 `False`。

```

1 s1 = set()
2 s2 = set("abc").clear()
3 print(s1 == s2)

```

第10题 执行下面 Python 代码后，输出的结果是 `{'x': 1, 'y': 2, 'z': None}`。

```

1 z = zip(['x', 'y', 'z'], [1, 2])
2 d = dict(z)
3 print(d)

```

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称：公共二进制位
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.1.1 题目描述

小红有 n 个非负整数 $a_1, a_2, \dots, a_n$ 。她将每个整数转换为二进制后，想知道有多少个二进制位在所有整数中均为 1。

二进制位从右向左编号为 $0, 1, 2, \dots$ 。若某个整数的二进制表示中没有第 k 位，则认为它的第 k 位为 0 。

请你求出满足条件的二进制位数量。

3.1.2 输入格式

- 第一行一个整数 n ，表示整数的个数。
- 第二行 n 个非负整数 $a_1, a_2, \dots, a_n$ 。

3.1.3 输出格式

输出一个整数，表示所有整数的二进制表示中均为 1 的二进制位数量。

3.1.4 样例

3.1.5 输入样例 1

```
1 3
2 13 7 15
```

3.1.6 输出样例 1

```
1 2
```

3.1.7 样例解释 1

三个整数的二进制表示分别为 $(1101)_2$ 、 $(0111)_2$ 和 $(1111)_2$ 。

其中第 0 位和第 2 位均为 1 ，因此答案为 2 。

3.1.8 数据范围

$2 \leq n \leq 20000$, $0 \leq a_i \leq 10^9$ 。

3.1.9 参考程序

```
1 # 接收数的数量
2 n = int(input())
3 # 输入各个数
4 aL = map(int, input().split())
5 # 所有位都是1
6 S = -1
7 # 挨个进行按位与，最后得到的S，某一位是1表明所有数这个位都是1
8 for a in aL:
9     S = S & a
10 ans = 0
11 # 对于S，不断右移，判断最后一位是否是1
12 while S > 0:
13     # S&1不为0就表明最后一位是1
14     if S & 1:
15         ans += 1
16     S >>= 1
17 print(ans)
```

3.2 编程题 2

- 试题名称：分割字符串
- 时间限制：1.0 s

- 内存限制：512.0 MB

3.2.1 题目描述

小明拿到了一段字符串文本，小明希望用下面的规则分割这个字符串：

1. 小明从左到右扫描字符串，遇到**第一个空格**后停止，并将字符串分割为左右两个部分（舍弃用于分割的这个空格）；
2. 对于上次分割后右侧的字符串，小明继续扫描，第一次遇到**连续两个空格**后停止，并将字符串分割为左右两个部分（舍弃用于分割的这两个空格）；
3. 对于上次分割后右侧的字符串，小明继续扫描，第一次遇到**连续三个空格**后停止，并将字符串分割为左右两个部分（舍弃用于分割的这三个空格）；
4. 以此类推，小明每分割一次，下一次分割就需要更长的连续空格；直到字符串无法再次分割。

最后输出每次分割的左侧的字符串，以及最后一次分割的右侧字符串。

如果字符串完全无法进行任何一次分割，直接输出其本身。

3.2.2 输入格式

输入一行，一个字符串。

3.2.3 输出格式

输出若干行。每行为分割后的一个字符串，注意依次输出：

第一次分割左侧的字符串，第二次分割左侧的字符串，...，最后一次分割左侧的字符串，最后一次分割右侧的字符串。

3.2.4 样例

3.2.5 输入样例 1

```
1 | sdfd 55 55 99312
```

3.2.6 输出样例 1

```
1 | sdfd
2 | 55 55
3 | 99312
```

3.2.7 样例解释 1

1. 第一次分割将字符串分割为 `sdfd` 和 `55 55 99312`；
2. 第二次分割将字符串分割为 `55 55` 和 `99312`。

对于第二次分割，注意 `55` 和 `55` 之间虽然有空格，但是第二次分割需要连续两个空格才行；

另外，`55 55` 和 `99312` 之间有三个空格，第二次分割只需要两个，所以会分割为 `55 55` 和 `99312`，也就是右侧会留下一个空格；因为只留下了一个空格，所以不再满足第三次分割要求。

3.2.8 输入样例 2

```
1 | AbCd 999
```

3.2.9 输出样例 2

```
1 | AbCd
2 |
3 | 999
```

3.2.10 样例解释 2

1. 第一次分割将字符串分割为 AbCd 和 999 ；
2. 第二次分割将字符串分割为 空字符串 和 999 。

由于 AbCd 和 999 之间有三个空格，第一次分割消耗掉一个，因此还能进行第二次分割，这个时候会分割出空字符串，因此输出中会输出一个空行（不能省略）。

3.2.11 数据范围

字符串长度不超过 1000，字符串中只包含大写字母、小写字母，数字，或空格，且输入字符串不会以空格结尾。

特别强调：输入仅一行，但可能包含若干空格

3.2.12 参考程序

```
1 | s = input()
2 | k = 1
3 | while True:
4 |     # 找到k个空格开始
5 |     p = s.find(' ' * k)
6 |     # 不存在就不管了
7 |     if p == -1:
8 |         break
9 |     # 把左边打印出来
10 |    print(s[:p])
11 |    # 右侧字符串
12 |    s = s[p + k:]
13 |    k += 1
14 |    # 把剩余字符串打印
15 |    print(s)
```