

**GESP****CCF 编程能力等级认证**  
Grade Examination of Software Programming

# Python 三级

2025 年 12 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	B	B	C	A	B	B	B	B	B	C	C	A	B	A	D

**第 1 题** 近日，空中客车公司表示，约6000架空客A320系列飞机需要紧急更换一种易受太阳辐射影响的飞行控制软件。空客表示，在对一起飞行事故进行分析后，表明强烈的太阳辐射可能会损坏飞行控制系统所需的关键数据，导致判断失误，进而可能引发飞行异常。在这里的飞行控制系统中，执行判断的部件最可能是下面的( )。

- A. 辐射传感器
- B. 处理器
- C. 内存单元
- D. 输出设备

**第 2 题** 小杨最近为了备考GESP，开始看网络上关于网络知识的视频。其中提到计算机网络系统有不同的划分标准，其中之一就是按照地理覆盖的范围来划分。那他平时上学所在的教学楼内的网络是一个( )。

- A. PAN
- B. LAN
- C. MAN
- D. WAN

**第 3 题** 下面选项中最大的数是？( )

- A.  $(42)_{10}$
- B.  $(2F)_{16}$
- C.  $(110111)_2$
- D.  $(41)_8$

**第 4 题** 在Python中，有一个整数变量  $x$ ，现需要将其二进制表示中的第三位（从右向左，最低位为第0位）设置为1，而其它位保持不变。下列哪段代码可以正确实现此功能？( )

- A.  $x = x | (1 << 3)$
- B.  $x = x & (1 << 3)$
- C.  $x = x ^ (1 << 2)$
- D.  $x = x | 0b100$

**第 5 题** 执行下面Python代码后，输出的结果是？( )

```
1 a = [1, 2]
2 b = a.copy()
3 a[0] = 3
4 print(b)
```

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

第6题 Python表达式 `[x % 3 for x in range(10) if x % 2 == 0]` 的值是?

- A. [0, 1, 0, 1, 0]
- B. [0, 2, 1, 0, 2]
- C. [0, 1, 2, 0, 1]
- D. [0, 2, 4, 6, 8]

第7题 已知 `lst = list(range(10))`，即 `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`。要获取切片 `[8, 6, 4, 2]`，应该使用的切片操作是?

- A. `lst[-2:2:-2]`
- B. `lst[8:1:-2]`
- C. `lst[8:2:-2]`
- D. `lst[-2:1:2]`

第8题 分析以下代码段，最终的输出结果是什么? ( )

```
1 config = {
2     'debug': True,
3     'port': 8080,
4     'debug': False,
5     'timeout': 30
6 }
7 print(config['debug'])
```

- A. True
- B. False
- C. 程序报错
- D. `{'debug': True, 'port': 8080, 'timeout': 30}`

第9题 合并两个字典 `d1 = {'a':1}` 和 `d2 = {'b':2}` 的正确方式是? ( )

- A. `d1 + d2`
- B. `d1.update(d2)`
- C. `d1.extend(d2)`
- D. `d1.append(d2)`

第10题 关于字符串的不可变性，下列说法正确的是? ( )

- A. 可以通过索引直接修改字符串中的某个字符

- B. 字符串的 `replace` 方法会直接修改原字符串
- C. 任何看似修改字符串内容的操作（如 `upper()`, `lower()` 等）都会返回一个新字符串，原始字符串不受影响。
- D. 字符串拼接操作 `+=` 会直接在原字符串末尾追加内容

**第 11 题** 对于字符串 `s = "Hello, Python"`，执行 `s.find("Java")` 和 `s.index("Java")` 的结果分别是? ( )

- A. -1, -1
- B. 程序报错, 程序报错
- C. -1, 程序报错
- D. 程序报错, -1

**第 12 题** 执行 `(1, 2) + (3, 4)` 的结果是? ( )

- A. (1, 2, 3, 4)
- B. (4, 3, 2, 1)
- C. (1, 3, 2, 4)
- D. 报错

**第 13 题** 在以下场景中，使用元组比使用列表更合适的是? ( )

- A. 存储一个需要频繁更新用户分数的排行榜。
- B. 表示一个二维平面上的点坐标  $(x, y)$ 。
- C. 缓存一批需要动态增删的临时数据。
- D. 存储一组程序运行时需要不断追加记录的日志信息。

**第 14 题** 表达式 `{x**2 for x in range(-2, 3)}` 的显示结果可能是? ( )

- A. {0, 1, 4}
- B. {4, 1, 0, 1, 4}
- C. {0, 1, 2, 4}
- D. {1, 4, 9}

**第 15 题** 现有集合 `s1 = {1, 2, 3}` 和 `s2 = {3, 4, 5}`，表达式 `(s1 - s2) | (s2 - s1)` 的结果是:

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

## 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** 鸿蒙是华为公司开发的一款操作系统，那么它能够将正确的源程序翻译成目标程序，并运行。 ( )

**第 2 题** 二进制数的奇偶性可以通过其最低位（最右边的位）来判断：最低位为0是偶数，为1是奇数。

第3题 在Python中，`print(~0)` 的输出结果仍是 0。

第4题 执行下面Python代码后，输出的结果是 `['apple', 'grape', 'banana', 'cherry', 'orange']`。

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

第5题 下面Python代码的作用是统计字符串 `s` 中每个字符出现的次数。

```
1 s = "abacab"
2 dict_count = {}
3 for char in s:
4     dict_count[char] = dict_count.get(char) + 1
5 print(dict_count)
```

第6题 执行下面Python代码后，输出的结果是 2。

```
1 s = "aaaaa"
2 print(s.count("aa"))
```

第7题 执行代码 `a = (1, 2, 3); a.sort()` 后，打印 `a` 会输出 `[1, 2, 3]`。

第8题 表达式 `{}` 在 Python 中可以创建一个空集合。

第9题 字符串可以通过 `list()` 函数直接转换为列表，但无法通过 `tuple()` 函数直接转换为元组。

第10题 以下代码是使用枚举算法，找出所有各位数字的立方和等于该数本身的三位数。

```
1 for num in range(100, 1000):
2     hundreds = num // 100
3     tens = (num // 10) % 10
4     units = num % 10
5     if hundreds**3 + tens**3 + units**3 == num:
6         print(num)
```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：密码强度
- 时间限制：1.0 s
- 内存限制：512.0 MB

##### 3.1.1 题目描述

小杨是学校网络安全小组的成员，今天他的任务是设计一个“密码强度检测器”，帮助同学们检查自己的密码是否足够安全。一个安全的密码需要满足以下条件：

- 密码至少包含8个字符（太短的密码容易被猜出来哦！）。
- 密码至少包含一个大写字母（A、B、C、...、Z 都可以）。
- 密码至少包含一个数字（0、1、2、3、...、9 都可以）。

例如：

- 密码 `PAs1s2an` 是安全密码（有8位、包含大写字母 `P`、`A` 和数字 `1`、`2`）。
- 密码 `ab1da3cd` 不是安全密码（没有大写字母）。
- 密码 `Paabdbcd` 不是安全密码（没有数字）。
- 密码 `Pa2` 不是安全密码（只有3位，太短了）。

### 3.1.2 输入格式

第一行一个正整数  $T$ ，代表需要安全检测的密码组数。

对于每组密码，一行包含一个字符串，代表需要安全检测的密码。

### 3.1.3 输出格式

对于每组密码，输出一行，如果满足强度要求输出 `Y`，否则输出 `N`。

### 3.1.4 样例

#### 3.1.4.1 输入样例

```

1 6
2 PAs1s2an
3 1a2bCql3
4 Pa12bsna
5 ab1da3cd
6 Paabdbcd
7 Pa2

```

#### 3.1.4.2 输出样例

```

1 Y
2 Y
3 Y
4 N
5 N
6 N

```

#### 3.1.4.3 样例解释

- 密码 `PAs1s2an` 是安全密码（有8位、包含大写字母 `P`、`A` 和数字 `1`、`2`）。
- 密码 `1a2bCql3` 是安全密码（有8位、包含大写字母 `C` 和数字 `1`、`2`、`3`）。
- 密码 `Pa12bsna` 是安全密码（有8位、包含大写字母 `P` 和数字 `1`、`2`）。
- 密码 `ab1da3cd` 不是安全密码（没有大写字母）。
- 密码 `Paabdbcd` 不是安全密码（没有数字）。
- 密码 `Pa2` 不是安全密码（只有3位，太短了）。

### 3.1.5 数据范围

对于所有测试点，保证  $1 \leq T \leq 100$ ，并且每组密码长度不超过 100 且至少为 1，每组密码仅由大小写字母和数字组成。

### 3.1.6 参考程序

```
1 # 通过int(input())获取输入数据的组数, 然后遍历每一条输入数据
2 for _ in range(int(input())):
3     # 接收当前的代码
4     password = input()
5     # 这个flag表示密码是否包含至少8位
6     flag_8 = len(password) >= 8
7     # 这个flag表示密码是否包含至少一个大写字母
8     flag_upper = False
9     for c in password:
10         if c.isupper():
11             flag_upper = True
12     # 这个flag表示密码是否包含至少一个数字
13     flag_digit = False
14     for c in password:
15         if c.isdigit():
16             flag_digit = True
17     if flag_8 and flag_upper and flag_digit:
18         print("Y")
19     else:
20         print("N")
```

## 3.2 编程题 2

- 试题名称: 小杨的智慧购物
- 时间限制: 1.0 s
- 内存限制: 512.0 MB

### 3.2.1 题目描述

小杨的班级要举办一个环保手工作品展览, 老师请小杨去文具店购买  $M$  种不同的文具 (例如: 铅笔、橡皮、尺子等)。

商店里共有  $N$  件文具, 每件文具都有一个种类编号 (从 1 到  $M$ ) 和价格。

小杨的预算有限, 他想了一个聪明的办法: 对于每种文具, 他只买最便宜的那一件 (如果同种文具有多件价格相同且都是最便宜的, 他只会购买其中的一件)。请你帮小杨计算出, 买齐这  $M$  种文具一共需要花费多少钱。

### 3.2.2 输入格式

第一行两个正整数  $M, N$ , 代表文具的种类数和总数。

之后  $N$  行, 每行两个正整数  $K_i$  和  $P_i$ , 分别代表第  $i$  件文具的种类编号和它的价格。数据保证每个种类至少有一件文具可供购买。

### 3.2.3 输出格式

输出一行, 代表购买文具的总价。

### 3.2.4 样例

#### 3.2.4.1 输入样例

```
1 2 5
2 1 1
3 1 2
4 1 1
5 2 3
```

### 3.2.4.2 输出样例

1	4
---	---

### 3.2.4.3 样例解释

文具清单如下：

- 文具 1：种类 1，价格 1
- 文具 2：种类 1，价格 2
- 文具 3：种类 1，价格 1
- 文具 4：种类 2，价格 3
- 文具 5：种类 2，价格 10

小杨的选择过程：对于种类 1：有三件商品，价格分别为 1, 2, 1。其中最便宜的价格是 1。对于种类 2：有两件商品，价格分别为 3, 10。其中最便宜的价格是 3。

计算总价：小杨购买这两类文具的总花费为  $1 + 3 = 4$ 。

### 3.2.5 数据范围

对于所有测试点，保证  $1 \leq M \leq N \leq 10^5, 1 \leq K_i \leq M, 1 \leq P_i \leq 10^3$ 。

### 3.2.6 参考程序

```

1 # 首先接收商品种类数量与商品总数
2 M, N = map(int, input().split())
3 # 记录各类商品的最低价格，初始化的时候先初始化为正无穷
4 min_price = [float('inf')] * (M + 1)
5
6 # 遍历每一个商品
7 for _ in range(N):
8     # 得到商品类别和商品的价格
9     K, P = map(int, input().split())
10    # 更新该类商品的最低价
11    min_price[K] = min(min_price[K], P)
12 # 将各类商品的最低价加和得到最后的结果
13 print(sum(min_price[1:M+1]))

```