

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

# Python 一级

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	A	A	B	A	C	A	B	B	B	A	D	B	B	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，结果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** 下面的Python代码在集成开发环境中运行时，提示有“invalid character”错误。可能的原因是( )：

```
1 a, b = 3, 4 #L1
2 print(a) #L2
3 print(b) #L3
```

- A. L1行代码中的逗号很可能是中文逗号，应该改为英文逗号。
- B. L1行应该分为两行，分别是 a = 3 和 b = 4。
- C. 代码运行前没有保存到文件。
- D. L2和L3不能分为两行，应合并为一行。

**第3题** 下面有关Python变量的说法，正确的是( )。

- A. 不可以用 for 作为变量名，因为 for 是Python的关键字（保留字）。
- B. \_tnt 不可以是变量名，因为变量名的第一个字符必须是英文字母。
- C. \_tnt\_ 不可以是变量名，因为最后一个字符容易与减号混淆。
- D. 可以用 print 作为变量名，因为 print 是关键字，但这不是好习惯，因为 print 有约定的功能与含义。

**第4题** Python表达式 2 + 3 \* 4 \*\* 2 的值为( )。

- A. 20
- B. 50
- C. 52
- D. 80

第5题 下面的Python代码执行后，其输出是()。

```
1 a = 3
2 b = a = 4
3 print(a, b)
```

- A. 4 4
- B. 3 3
- C. 3 4
- D. 4 3

第6题 下面的Python代码执行时如果先输入 10 回车后输入 20 并回车，其输出是()。

```
1 N = input("第一个数:")
2 M = input("第二个数:")
3 print(f"int(N+M)={int(N+M)}")
```

- A. 30=30
- B. 1020=1020
- C. int(N+M)=1020
- D. 错误提示

第7题 某个整数很长很长，形如：1232123212321……，其规律是从1开始逐一升高到3然后逐一降低到1，然后又逐一升高到3，一直到很长很长。编写程序，判断从左开始第N位数是几？假设从左到右第1位为1。在横线处应该填入的代码是()。

```
1 N = int(input("请输入编号: "))
2 M = -----
3
4 if M != 0:
5     print(M)
6 else:
7     print(2)
```

- A. N % 4
- B. N // 4
- C. N % 3
- D. N // 3

第8题 下面Python代码执行后的输出是()。

```
1 tnt = 0
2 for i in range(100):
3     tnt += 1
4 print(tnt, i)
```

- A. 99 99
- B. 100 99
- C. 99 100
- D. 100 100

第9题 有关下面Python代码的说法，错误的是()。

```
1 | tnt = 0
2 | for i in range(1, 10, 2):
3 |     tnt += i
4 | print(tnt)
```

- A. range(1,10,2) 改为 range(1,11,2) 结果相同。
- B. range(1,10,2) 改为 range(0,10,2) 结果相同。
- C. tnt += i 与 tnt = tnt + i 效果相同。
- D. tnt += i 与 tnt = i + tnt 效果相同。

第 10 题 下面Python代码执行后输出是()。

```
1 | for i in range(10, 100, 10):
2 |     if i % 10 == 0:
3 |         continue
4 |     print(i, end = "#")
```

- A. 10#20#30#40#50#60#70#80#90#
- B. 90#
- C. 90
- D. 没有输出

第 11 题 两个正整数，只要不相等，就一直进行如下操作：最大数减去最小数得到一个值，该值和两个数的最小数构成新的两个正整数，重复操作，直到两个数相等，此时输出该数。下面的Python代码用于实现该操作，横线处应该填写的代码是()。

```
1 | N = int(input())
2 | M = int(input())
3 |
4 | while N != M:
5 |     if N > M:
6 |         -----
7 |     else:
8 |         -----
9 |
10 | print(N)
```

- A.

```
1 | N = N - M
2 | M = M - N
```

- B.

```
1 | M = M - N
2 | N = N - M
```

- C.

```
1 | M = N - M
2 | N = M - N
```

- D.

```
1 | N, M = M, N
2 | M, N = N, M
```

**第12题** 如果一个正整数能被3整除，或者某一位能被3整除，则称之为“漂亮数”，如12是漂亮数因为能被3整除，311是漂亮数因为含有3，22不是漂亮数因为不能被3整除也不含有3。下面的Python用于判断正整数  $N$  是否为漂亮数，横线处应该填入的代码是()。

```
1 N = int(input())
2
3 Flag = "非漂亮数"
4 if N % 3 == 0:
5     Flag = "漂亮数"
6 else:
7     while N != 0:
8         if _____:
9             Flag = "漂亮数"
10            break
11         N //= 10
12 print(Flag)
```

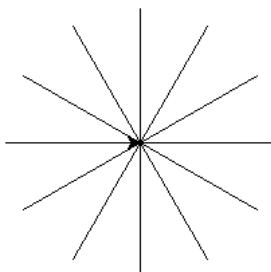
- A.  $N \% 10 == 0$
- B.  $N \% 3 \% 10 == 0$
- C.  $N \% 10 \% 3$
- D.  $N \% 10 \% 3 == 0$

**第13题** 下面的Python代码执行后海龟最终朝向是()。 (默认初始朝向东，即屏幕右侧)

```
1 import turtle
2 turtle.forward(100)
3 turtle.right(90)
4 turtle.forward(50)
```

- A. 东 ( $0^\circ$ )
- B. 南 ( $270^\circ$  或  $-90^\circ$ )
- C. 西 ( $180^\circ$ )
- D. 北 ( $90^\circ$ )

**第14题** 为在Python Turtle中输出如下图形，代码横线处应填入()。

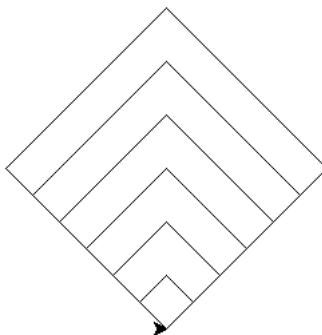


```
1 import turtle
2 for i in range(12):
3     turtle.forward(100)
4     _____
5     turtle.left(30)
```

- A. `turtle.home()`
- B. `turtle.goto(0,0)`
- C. `turtle.reset()`

D. `turtle.home(0,0)`

第15题 为在Python Turtle中输出如下图形，代码横线处应填入()。



```
1 import turtle
2 for i in range(6):
3     turtle.circle(-----)
```

A. `20 * i`

B. `20 * (i + 1)`

C. `20 * i, steps = 4`

D. `20 * (i + 1), steps = 4`

## 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 Python表达式 `10 ** 0.5` 和 `10 ** 1 / 2` 的结果并不相同。 ( )

第3题 下面Python代码执行后将输出 1。 ( )

```
1 for i in range(10):
2     if i % 3 == 0:
3         continue
4     else:
5         break
6 print(i)
```

第4题 下面Python代码中的 `range(-10)` 改为 `range(10)` 执行结果相同。 ( )

```
1 tnt = 0
2 for i in range(-10):
3     if i < 0:
4         i = -i
5     tnt += i
6 print(tnt)
```

第5题 下面Python代码执行后将输出 99，因为 `range(100)` 不包括 100，而是到 99。 ( )

```
1 | cnt = 0
2 | for i in range(100):
3 |     cnt += 1
4 | print(cnt)
```

**第6题** 下面的Python代码能用于判断输入的正整数是否为对称数。所谓对称数是指从左到右或从右到左读该数，其值相同，如 121 或 414 等是对称数，而 123 不是对称数。 ( )

```
1 | n = int(input("请输入正整数:"))
2 |
3 | new_number = 0
4 | while n != 0:
5 |     new_number = new_number * 10 + n % 10
6 |     n //= 10
7 |
8 | if n == new_number:
9 |     print("对称数")
10 | else:
11 |     print("非对称数")
```

**第7题** 下面的Python代码执行后将输出 -100 。 ( )

```
1 | tnt = 0
2 | for i in range(-100,100):
3 |     tnt += i
4 | print(tnt)
```

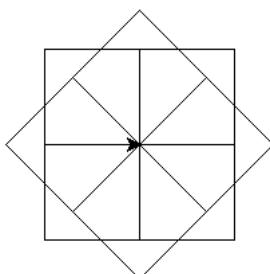
**第8题** 下面Python代码执行后，其输出是 -50 。 ( )

```
1 | tnt = 0
2 | flag = 1
3 | for i in range(1,100 + 1):
4 |     tnt += i * flag
5 |     flag = -flag
6 | print(tnt)
```

**第9题** 下面的Python代码执行后，将画出横平竖直(两条边水平，另两条边与之垂直，且首尾相接)的正方形。 ( )

```
1 | import turtle
2 | turtle.right(45)
3 | turtle.circle(45, steps = 4)
```

**第10题** 下面的Python代码执行后将绘制出如下图形。 ( )



```
1 | import turtle
2 | for i in range(8):
3 |     turtle.right(45)
4 |     turtle.circle(50,steps = 4)
```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：小杨的爱心快递
- 时间限制：1.0 s
- 内存限制：512.0 MB

##### 3.1.1 题目描述

小杨是“爱心社区”的小志愿者，每周他都会帮助邻居们寄送捐赠给山区小学的文具和书籍。快递公司为了支持公益行动，制定了特殊的运费规则，鼓励大家合理包装：

假设快递的体积为  $V$ ，重量为  $G$ 。

- 按体积计算：运费按体积计算，公式是  $0.5 \times V$  元。
- 按重量计算：为了鼓励减轻包裹重量，规则是：当重量小于 300 克，即  $G < 300$  时，运费为  $M$  元；当重量达到或超过 300 克，即  $G \geq 300$  时，运费为  $N$  元。

快递公司的叔叔说：“我们应该选择最公平合理的计费方式。”

所以，最终的运费会取按体积计算和按重量计算这两种方式中价格较低的那个，这样对寄件人最公道。

##### 3.1.2 输入格式

四行，每行一个一位小数的浮点数，分别代表，快递的体积  $V$ ，快递的重量  $G$ ，第一档重量运费  $M$ ，第二档重量运费  $N$ 。

##### 3.1.3 输出格式

一行一个一位小数，代表实际快递运费。

##### 3.1.4 样例

###### 3.1.4.1 输入样例 1

1	100.4
2	300.2
3	60.6
4	70.5

###### 3.1.4.2 输出样例 1

1	50.2
---	------

###### 3.1.4.3 输入样例 2

1	99.8
2	200.9
3	60.2
4	70.1

### 3.1.4.4 输出样例 2

```
1 | 49.9
```

### 3.1.4.5 样例解释

- 对于样例 1: 快递体积  $V$  为 100.4, 重量  $G$  为 300.2, 第一档重量运费  $M$  为 60.6, 第二档重量运费  $N$  为 70.5。
  - 若按体积收费, 费用为  $0.5 \times 100.4 = 50.2$ ;
  - 若按重量收费, 由于重量  $G = 300.2$  超过 300, 费用为 70.5。

输出价格较低的 50.2。

- 对于样例 2: 快递体积  $V$  为 99.8, 重量  $G$  为 200.9, 第一档重量运费  $M$  为 60.2, 第二档重量运费  $N$  为 70.1。
  - 若按体积收费, 费用为  $0.5 \times 99.8 = 49.9$ ;
  - 若按重量收费, 由于重量  $G = 200.9$  小于 300, 费用为 60.2。

输出价格较低的 49.9。

### 3.1.5 数据范围

对于所有测试点, 保证给出的浮点数不超过 1000。

### 3.1.6 参考程序

```
1 V = float(input()) #输入体积V
2 G = float(input()) #输入重量W
3 M = float(input()) #第一档重量运费M
4 N = float(input()) #第二档重量运费 N
5
6 #按体积计费时的费用总额
7 p_v = V / 2
8
9 #按重量档次计费计算
10 if G < 300:
11     p_g = M #如果小于300, 则按此计费
12 else:
13     p_g = N #按第二档计费
14
15 #比较体积计费和重量计费, 选择费用少的计费方式
16 if p_v < p_g:
17     print(f"{p_v:.1f}")
18 else:
19     print(f"{p_g:.1f}")
```

## 3.2 编程题 2

- 试题名称: 手机电量显示
- 时间限制: 1.0 s
- 内存限制: 512.0 MB

### 3.2.1 题目描述

小杨的手机就像一个聪明的小助手, 当电量变化时, 它会用不同的方式来提醒我们, 假设当前的电量百分比为  $P$ :

- 当电量非常低 (不超过 10, 即  $P \leq 10$ ) , 它会显示一个大写字母 R, 就像在说: “快给我充电吧! (Red警告色) ”
- 当电量有点低 (超过 10 但不超过 20, 即  $10 < P \leq 20$ ) , 它会显示一个大写字母 L, 意思是“电量有点Low

啦！”

- 当电量比较充足（超过 20，即  $P > 20$ ），它就会直接显示具体的数字，比如直接显示 50，表示还有 50 的电量。

### 3.2.2 输入格式

第一行一个正整数  $T$ ，代表数据组数。

对于每组数据，一行包含一个正整数  $P$ ，代表手机电量百分比。

### 3.2.3 输出格式

对于每组数据，输出一行，代表当前手机显示的电量信息。

### 3.2.4 样例

#### 3.2.4.1 输入样例

1	5
2	10
3	1
4	20
5	99
6	19

#### 3.2.4.2 输出样例

1	R
2	R
3	L
4	99
5	L

#### 3.2.4.3 样例解释

输入样例 1 共有五组数据：

- 第一组数据手机电量  $P = 10$ ，满足  $P \leq 10$ ，电量非常低，显示 R。
- 第二组数据手机电量  $P = 1$ ，满足  $P \leq 10$ ，电量非常低，显示 R。
- 第三组数据手机电量  $P = 20$ ，满足  $10 < P \leq 20$ ，电量有点低，显示 L。
- 第四组数据手机电量  $P = 99$ ，满足  $P > 20$ ，电量比较充足，直接显示具体的数字 99。
- 第五组数据手机电量  $P = 19$ ，满足  $10 < P \leq 20$ ，电量有点低，显示 L。

### 3.2.5 数据范围

对于所有测试点，保证  $1 \leq T \leq 20, 1 \leq P \leq 100$ 。

### 3.2.6 参考程序

```
1 # 读取测试用例的数量 T
2 line = input() # 读入数据行的数量, line的数据类型为字符串
3 T = int(line) # 将line转换为整数, 表示输入数据行的数量
4
5 # 循环 T 次
6 # 循环变量仅用于控制循环次数
7 for _ in range(T):
8     # 读取 P
9     P = int(input())
10
11     # 判断逻辑
12     if P <= 10:
13         print("R") #电量小于10%时
14     elif P <= 20: #电量小于20%时
15         print("L")
16     else:
17         print(P) #电量超过20%时
```