



# Python 二级

2025 年 03 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	D	C	B	A	A	D	A	D	A	C	B	D	C	D	B

第 1 题 2025年春节有两件轰动全球的事件，一个是DeepSeek横空出世，另一个是贺岁片《哪吒2》票房惊人，入了全球票房榜。下面关于DeepSeek与《哪吒2》的描述成立的是( )。

- ☐ A. 《哪吒2》是一款新型操作系统
- ☐ B. DeepSeek是深海钻探软件
- ☐ C. 《哪吒2》可以生成新的软件
- ☐ D. DeepSeek可以根据《哪吒2》的场景生成剧情脚本

第 2 题 对整型变量N，如果它能够同时被3和5整出，则输出 N是含有至少两个质因数。如果用流程图来描述处理过程，则输出语句应该在何种图形框中 ( )。

- ☐ A. 圆形框
- ☐ B. 椭圆形框
- ☐ C. 平行四边形框
- ☐ D. 菱形框

第 3 题 Python语句 `print(3+3**(2-2)*3)` 执行后输出的值是 ( )。

- ☐ A. 11
- ☐ B. 6
- ☐ C. 4
- ☐ D. 3

第 4 题 下面Python代码执行，其输出是( )。

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

- ☐ A. 3 4

- ☐ B. 3 3
- ☐ C. 4 4
- ☐ D. 4 3

第5题 求三色彩球的颜色。有数量无限的红(Red)绿(Green)蓝(Blue)三种彩球排成一行，每组先为5个红色球，随后3个绿色，最后为2个蓝色。每个球都有编号，从左到右依次为1,2,3……。输入整数代表编号，求该编号球的颜色。下面是Python代码的实现，正确说法是( )。

```
1 N = int(input())
2 remainder = N % 10 #remainder变量保存余数
3
4 if 1 <= remainder <= 5:
5     print("Red")
6 elif 6 <= remainder <= 8:
7     print("Green")
8 elif remainder == 9 or remainder == 0:
9     print("Blue")
```

- ☐ A. 将 `elif remainder == 9 or remainder == 0:` 修改为 `else` 效果相同
- ☐ B. 将 `if 1 <= remainder <= 5:` 修改为 `if remainder <= 5:` 效果相同
- ☐ C. `elif 6 <= remainder <= 8:` 写法错误，应修改为 `elif 6 <= remainder and remainder <= 8:`
- ☐ D. 根据题意 `remainder = N % 10` 应修改为 `remainder = N // 10`

第6题 下面Python代码执行后其输出是( )。

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

- ☐ A. 18
- ☐ B. 17
- ☐ C. 16
- ☐ D. 14

第7题 下面Python代码执行后输出是( )。

```
1 for i in range(10, 0, -2):
2     break
3 print(i)
```

- ☐ A. 10
- ☐ B. 8
- ☐ C. 0

☐ D. 因为循环执行时会执行break语句而终止循环，因此i的值不确定

第8题 下面Python代码执行后输出是（ ）。

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

☐ A. 0#0#0#0#0#0#

☐ B. 0#0#0#0#0#0#0#1#

☐ C. 0#0#0#0#1#

☐ D. 0#0#0#0#0#0#1#

第9题 下面Python代码执行后的输出是（ ）。

```
1 for i in range(5):
2     for j in range(i,0,-1):
3         print(j,end="-")
```

☐ A. 1-2-1-3-2-1-4-3-2-1-

☐ B. 1-2-1-3-2-1-4-3-2-1

☐ C. 0-0-1-0-1-2-0-1-2-3-

☐ D. 0-0-1-0-1-2-0-1-2-3

第10题 下面Python代码执行后，将输出能被2整除且除以7余数为2的数。下列选项不能实现的是（ ）。

```
1 for i in range(100):
2     if _____:
3         print(i)
```

☐ A.  $i \% 2 == 0$  and  $i \% 7 == 2$

☐ B.  $\text{not}(i \% 2)$  and  $i \% 7 == 2$

☐ C.  $\text{not}(i \% 2)$  and  $\text{not}(i \% 7)$

☐ D.  $i \% 2 != 1$  and  $i \% 7 == 2$

第11题 下面Python代码用于求正整数1到N之间含有3的数的个数，横线处应填入代码先后是（ ）。

```

1 cnt = 0
2 N = int(input("请输入正整数: "))
3 for i in range(1,N):
4     while i != 0:
5         if i % 10 == 3:
6             cnt +=1
7             _____
8         else:
9             _____
10 print(cnt)

```

☐ A.

```

1 continue
2 i = i // 10

```

☐ B.

```

1 break
2 i //= 10

```

☐ C.

```

1 continue
2 i %= 10

```

☐ D.

```

1 break
2 i %= 10

```

第 12 题 下面Python代码用于实现九九乘法表，代码中{i\*j}与等号之间有一个空格。代码执行后效果是（ ）。

```

1 for i in range(1, 10):
2     for j in range(1, i + 1):
3         print(f"{j}*{i}= {i*j}", end=" ")
4     print()

```

☐ A.

```

1*1=1
1*2=2  2*2=4
1*3=3  2*3=6  3*3=9
1*4=4  2*4=8  3*4=12  4*4=16
1*5=5  2*5=10  3*5=15  4*5=20  5*5=25
1*6=6  2*6=12  3*6=18  4*6=24  5*6=30  6*6=36
1*7=7  2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49
1*8=8  2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64
1*9=9  2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81

```

☐ B.

```
1*1= 1
1*2= 2 2*2= 4
1*3= 3 2*3= 6 3*3= 9
1*4= 4 2*4= 8 3*4=12 4*4=16
1*5= 5 2*5=10 3*5=15 4*5=20 5*5=25
1*6= 6 2*6=12 3*6=18 4*6=24 5*6=30 6*6=36
1*7= 7 2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49
1*8= 8 2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64
1*9= 9 2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81
```

☐ C.

```
1*1=1
1*2=2 2*2=4
1*3=3 2*3=6 3*3=9
1*4=4 2*4=8 3*4=12 4*4=16
1*5=5 2*5=10 3*5=15 4*5=20 5*5=25
1*6=6 2*6=12 3*6=18 4*6=24 5*6=30 6*6=36
1*7=7 2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49
1*8=8 2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64
1*9=9 2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81
```

☐ D.

```
1*1= 1
1*2= 2 2*2= 4
1*3= 3 2*3= 6 3*3= 9
1*4= 4 2*4= 8 3*4= 12 4*4= 16
1*5= 5 2*5= 10 3*5= 15 4*5= 20 5*5= 25
1*6= 6 2*6= 12 3*6= 18 4*6= 24 5*6= 30 6*6= 36
1*7= 7 2*7= 14 3*7= 21 4*7= 28 5*7= 35 6*7= 42 7*7= 49
1*8= 8 2*8= 16 3*8= 24 4*8= 32 5*8= 40 6*8= 48 7*8= 56 8*8= 64
1*9= 9 2*9= 18 3*9= 27 4*9= 36 5*9= 45 6*9= 54 7*9= 63 8*9= 72 9*9= 81
```

第13题 在数学中 $N!$ 表示 $N$ 的阶乘，即1到 $N$ 的乘积，如 $3!=1*2*3$ ，且 $0!=1$ 。下面的两段Python代码用于求1- $N$ 的阶乘之和，如 $N$ 为3，则是 $1!+2!+3!$ 。选项中的说法，正确的是（ ）。

```
1 #实现1
2 N = int(input())
3 tnt, last = 0, 1
4 for i in range(1,N + 1):
5     last *= i
6     tnt += last
7 print(tnt)
```

```
1 #实现2
2 N = int(input())
3 tnt = 0
4 for i in range(1, N + 1):
5     tmp = 1
6     for j in range(1,i + 1):
7         tmp *= j
8     tnt += tmp
9 print(tnt)
```

☐ A. 虽然实现1的代码短小，但效率并不高

- ☐ B. 实现2的代码效率更高，且更易于理解
- ☐ C. 实现1因为应用了前项计算结果，计算量更小，因此效率高
- ☐ D. 两种实现，效率几乎一致

**第 14 题** 哥德巴赫猜想是指大于2的偶数都可以分解为两个质数之和，下面的代码用于验证4-1000之内的偶数能否分解为两个质数之和。下面Python代码中假设isPrime()是已经定义好用于判断正整数N是否为质数。对该段代码，错误的说法是（ ）。

```
1 for i in range(4,1000,2):
2     for j in range(2,i):
3         if isPrime(j) and isPrime(i-j):
4             print(f"{i}={j}+{i-j}")
5             break
```

- ☐ A. 将代码 isPrime(j) and isPrime(i-j) 修改为  
isPrime(j) == True and isPrime(i-j) == True 效果相同
- ☐ B. 代码执行后，输出的一对质数，一定是小的数在前
- ☐ C. 即便将range(4,1000,2)中的1000修改为很大的整数，也不能说从数学上证明了哥德巴赫猜想
- ☐ D. 根据题意，break语句应该与if对齐

**第 15 题** 下面Python代码实现输出如下图形，相关说法正确的是（ ）。

请输入层数：10

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

```
1 N = int(input("请输入层数: "))
2
3 last = 1
4 for i in range(1,N):
5     for j in range(1,i + 1):
6         if last > 9:
7             last = 1
8
9         print(last,end = " ")
10        last += 1
11
12    print()
```

- ☐ A. 最后一行的print()有错，应该修改为print(""), print()函数不能没有参数
- ☐ B. last += 1修改为last = last + 1执行效果相同
- ☐ C. 代码中的range(1, i + 1)应修改为range(1, i)
- ☐ D. 外层for循环前的last = 1修改为last = 0执行效果相同

## 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 C++、Python都是高级编程语言，它们每条语句的执行最终都要通过机器指令来完成。（ ）

第 2 题 在Python代码中，假设N为正整数，则  $N - N // 10 * 10$  与  $N \% 10$  都将获得N的个位数。（ ）

第 3 题 在Python语句 `print(10 <= N <= 12)` 中，假设N为12，则其输出为True。原因是执行  $10 <= N$  后其值为True，True 与 12 相比仍然是 True。（ ）

第 4 题 Python表达式  $N ** 0.5 ** 2 == N$  中的N如果为正整数，则表达式的值为True，相当于开平方后平方是本身。（ ）

第 5 题 下面Python执行后将输出 $3*2=6$ 。（ ）

```
1 a, b = 2, 3
2 a, b = b, a
3 print(f"{a}*{b}={a*b}")
```

第 6 题 下面Python代码执行后将输出10。（ ）

```
1 for i in range(10):
2     continue
3 print(i)
```

第 7 题 下面Python代码执行后将输出1。（ ）

```
1 for i in range(1, 10):
2     break
3     continue
4 print(i)
```

第 8 题 下面的Python代码执行后将输出10行“OK”。（ ）

```
1 for i in range(5):
2     for j in range(i):
3         print("OK")
```

第 9 题 将下面Python代码中的`range(1, 5)`调整为`range(5)`输出结果相同。（ ）

```
1 tnt = 0
2 for i in range(1, 5):
3     tnt += i
4 print(tnt)
```

第 10 题 下面Python代码执行后将输出0123。（ ）

```
1 for i in range(5):
2     for i in range(i):
3         continue
4     print(i,end="")
```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 时间限制：1.0 s
- 内存限制：512.0 MB

##### 3.1.1 等差矩阵

##### 3.1.2 题目描述

小 A 想构造一个  $n$  行  $m$  列的矩阵，使得矩阵的每一行与每一列均是等差数列。小 A 发现，在矩阵的第  $i$  行第  $j$  列填入整数  $i \times j$ ，得到的矩阵能满足要求。你能帮小 A 输出这个矩阵吗？

##### 3.1.3 输入格式

一行，两个正整数  $n, m$ 。

##### 3.1.4 输出格式

共  $n$  行，每行  $m$  个由空格分割的整数，表示小 A 需要构造的矩阵。

##### 3.1.5 样例

###### 3.1.5.1 输入样例 1

```
1 | 3 4
```

###### 3.1.5.2 输出样例 1

```
1 | 1 2 3 4
2 | 2 4 6 8
3 | 3 6 9 12
```

##### 3.1.6 数据范围

对于所有测试点，保证  $1 \leq n \leq 50$ ， $1 \leq m \leq 50$ 。

##### 3.1.7 参考程序

```
1 | n, m = map(int, input().split())
2 | for i in range(1, n + 1):
3 |     print(*[i * j for j in range(1, m + 1)])
```

#### 3.2 编程题 2

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



### 3.2.8 题面描述

假设现在是  $y$  年  $m$  月  $d$  日  $h$  时而  $k$  小时后是  $y'$  年  $m'$  月  $d'$  日  $h'$  时，对于给定的  $y, m, d, h, k$ ，小杨想请你帮他计算出对应的  $y', m', d', h'$  是多少。

### 3.2.9 输入格式

输入包含五行，每行一个正整数，分别代表  $y, m, d, h, k$ 。

### 3.2.10 输出格式

输出四个正整数，代表  $y', m', d', h'$ 。

### 3.2.11 样例

```
1 2008
2 2
3 28
4 23
5 1
```

```
1 2008 2 29 0
```

### 3.2.12 数据范围

对于全部数据，保证有  $2000 \leq y \leq 3000, 1 \leq m \leq 12, 1 \leq d \leq 31, 0 \leq h \leq 23, 1 \leq k \leq 24$ 。数据保证为合法时间。

### 3.2.13 提示

闰年判断规则

- 普通闰年：年份能被 4 整除，但不能被 100 整除。
- 世纪闰年：年份能被 400 整除。

满足以上任意一条规则的年份就是闰年，否则是平年。

### 3.2.14 参考程序

```
1 y = int(input())
2 m = int(input())
3 d = int(input())
4 h = int(input())
5 k = int(input())
6
7 h += k
8
9 if h >= 24:
10     h -= 24
11     d += 1
12     days = 0
13     if m == 1 or m == 3 or m == 5 or m == 7 or m == 8 or m == 10 or m == 12:
14         days = 31
15     elif m == 4 or m == 6 or m == 9 or m == 11:
16         days = 30
17     elif m == 2:
18         if (y % 4 == 0 and y % 100 != 0) or (y % 400 == 0):
```

```
19         days= 29
20     else:
21         days= 28
22     if d > days:
23         d -= days
24         m += 1
25         if m > 12:
26             m = 1
27             y += 1
28 print(y,end=' ')
29 print(m,end=' ')
30 print(d,end=' ')
31 print(h)
```