



# Python 二级

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

第 1 题 人工智能现在非常火，小杨就想多了解一下，其中就经常听人提到“大模型”。那么请问这里说的“大模型”最贴切是指（ ）。

- ☐ A. 大电脑模型
- ☐ B. 大规模智能
- ☐ C. 智能的单位
- ☐ D. 大语言模型

第 2 题 在TCP协议中，完成连接建立需要通过（ ）握手。

- ☐ A. 一次
- ☐ B. 二次
- ☐ C. 三次
- ☐ D. 四次

第 3 题 下面的Python代码用于输入姓名，然后输出姓名，正确的说法是（ ）。

```
1 XingMing = input("请输入您的姓名：")
2 print(XingMing)
```

- ☐ A. XingMing 是汉语拼音，不能作为变量名称
- ☐ B. 可以将 XingMing 改为 Xing Ming
- ☐ C. 可以将 XingMing 改为 xingming
- ☐ D. 可以将 XingMing 改为 Xing-Ming

第 4 题 下面Python代码用于获得正整数N的第M位数，如N等于1234，M等于2，则输出3。此题假设M的值大于等于1小于等于N的位数。横线处应填入的代码是（ ）。

```
1 N = int(input("请输入一个正整数："))
2 M = int(input("请输入从右到左取第几位数："))
3
4 div = 10 ** (M - 1)
5
6 print(_____)
```

- ☐ A.  $N \% div // 10$

- ☐ B. `N // div // 10`
- ☐ C. `N % div % 10`
- ☐ D. `N // div % 10`

第5题 下面Python代码执行，其输出是( )。

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

- ☐ A. `False 4 4`
- ☐ B. `3 3 3`
- ☐ C. `4 4 4`
- ☐ D. `3 4 False`

第6题 某种编号的规则是“XX-Y”，其中XX从00到11，Y从0到9，X和Y均为10进制。第一个编号是00-0，第二个编号是01-1，第12个编号11-1，第13个编号00-2，即其编码规则是XX和Y同时增1，到XX到11时下一个变为00，Y到9时，下一个变为0。下面的Python代码用于生成第N个编号，横线处应填上的代码是( )。

```
1 | N = int(input("请输入编号位置: "))
2 |
3 | part1 = N % -----
4 | part2 = N % -----
5 |
6 | if part1 < 10:
7 |     print(f"0{part1}-{part2}")
8 | else:
9 |     print(f"{part1}-{part2}")
```

- ☐ A. `12 10`
- ☐ B. `10 10`
- ☐ C. `11 9`
- ☐ D. `9 9`

第7题 下面的Python代码执行后，将输出( )。

```
1 | for i in range(-2, 2):
2 |     if i:
3 |         print(i, end = "#")
```

- ☐ A. `-2#-1#1#`
- ☐ B. `-2#-1#0#1#`
- ☐ C. `-2#-1#1#2#`
- ☐ D. `-2#-1#0#1#2#`

第8题 下面的Python代码执行后其输出是( )。

```
1 | cnt = 0
2 | for i in range(-10,10):
3 |     for j in range(i):
4 |         cnt += 1
5 | print(cnt)
```

- ☐ A. `145`

- ☐ B. 125
- ☐ C. 55
- ☐ D. 45

第9题 下面Python代码执行后其输出是( )。

```
1 for i in range(1, 12):
2     if i % 2 == 0:
3         continue
4
5     for j in range(i):
6         if i * j % 2:
7             break
8     else:
9         print(i * j)
```

- ☐ A. 99
- ☐ B. 11
- ☐ C. 0
- ☐ D. 没有输出

第10题 阅读下面的Python代码，正确的是( )。

```
1 a = int(input())
2 b = int(input())
3
4 while b != 0:
5     remainder = a % b
6     a = b
7     b = remainder
8
9 print(a)
```

- ☐ A. b 不能为 0，因为 a % b 将导致错误
- ☐ B. a 必须小于 b，否则 a % b 将导致错误
- ☐ C. a 和 b 都必须为正整数，否则 a % b 将导致错误
- ☐ D. 如果 a 为 0，则不管 b 为多少，输出都将是 b

第11题 下面Python代码执行后输出是( )。

```
1 num = 0
2 while num <= 5:
3     num += 1
4     if num == 3:
5         continue
6     print(num, end='#')
```

- ☐ A. 1#2#4#5#6#
- ☐ B. 1#2#4#5#6
- ☐ C. 1#2#3#4#5#6#
- ☐ D. 1#2#3#4#5#6

第12题 下面Python代码用于记录多个输入数中的最大数和最小数（输入 -999 则输入结束），相关说法错误的是( )。

```

1 now_num = int(input())
2 min_num = max_num = now_num
3
4 while now_num != -999:
5
6     if max_num < now_num:
7         max_num = now_num
8
9     if min_num > now_num:
10        min_num = now_num
11
12    now_num = int(input())
13
14 print(min_num, max_num)

```

- ☐ A. 程序运行时如果第一个数输入 -999，则输出将是 -999 -999
- ☐ B. 程序输入过程中，如果输入的的第一个数不是 -999，且如果待输入的实际数据中没有 -999【即：-999 仅作为结束程序的标志数据】，则程序能求出最大数和最小数
- ☐ C. 如果用于输入考试成绩，即成绩中不可能有 -999，则程序能求出最高成绩和最低成绩
- ☐ D. 可以将 now\_num = int(input()) 移动到 while now\_num != -999: 之后

**第 13 题** 下面Python代码执行后输出与 5 有关数的数量。与5有关的数定义为含有 5 或者能被 5 整除的数。相关说法正确的是（ ）。

```

1 cnt = 0
2 for i in range(1,1000):
3     if i % 5 == 0:
4         cnt += 1
5         continue
6     j = i
7     while j > 0:
8         if j % 10 == 5:
9             cnt += 1
10            break
11        else:
12            j //= 10
13 print(cnt)

```

- ☐ A. 删除代码中 continue 不影响程序执行结果
- ☐ B. 删除代码中 break 不影响程序执行结果
- ☐ C. 删除 j = i 并将 while 循环的j修改为i，不影响程序执行结果
- ☐ D. 将 while j > 0 修正为 while j>= 0 不影响程序执行的结果

**第 14 题** 下面Python代码实现输出如下图形，应该在横线处填入的代码是（ ）【注：字符串乘以一个正整数，相当于相同字符重复】。

输入行数量：10  
输入每行字符数量：12

```

-----**-----
-----**-----
-----**-----
-----**-----
*****
*****
-----**-----
-----**-----
-----**-----
-----**-----

```

```

1 n = int(input("输入行数量: "))

```

```

2 m = int(input("输入每行字符数量: "))
3
4 # 计算行和列的中间位置
5 mid_row = n // 2
6 mid_col = m // 2
7
8 for i in range(n):
9     # 判断当前行是否为中间行 (或中间两行)
10    if -----:
11        print('*' * m)
12    else:
13        # 根据列数奇偶性构建行字符串
14        if m % 2 == 1:
15            # 奇数列: 左侧- + 中间* + 右侧-
16            left = '-' * mid_col
17            center = '*'
18            right = '-' * mid_col
19            print(left + center + right)
20        else:
21            # 偶数列: 左侧- + 中间** + 右侧-
22            left = '-' * (mid_col - 1)
23            center = '**'
24            right = '-' * (mid_col - 1)
25            print(left + center + right)

```

- ☐ A.  $(n \% 2 == 1 \text{ and } i == \text{mid\_row}) \text{ or } (n \% 2 == 0 \text{ and } (i == \text{mid\_row} - 1 \text{ or } i == \text{mid\_row}))$
- ☐ B.  $(n \% 2 == 0 \text{ and } i == \text{mid\_row}) \text{ or } (n \% 2 == 1 \text{ and } (i == \text{mid\_row} - 1 \text{ or } i == \text{mid\_row}))$
- ☐ C.  $(n \% 2 == 1 \text{ and } i == \text{mid\_row}) \text{ or } (n \% 2 == 0 \text{ and } (i == \text{mid\_row} \text{ or } i == \text{mid\_row} + 1))$
- ☐ D.  $(n \% 2 == 1 \text{ and } i == \text{mid\_row}) \text{ or } (n \% 2 == 0 \text{ and } i == \text{mid\_row} - 1) \text{ and } (n \% 2 == 0 \text{ and } i == \text{mid\_row})$

第 15 题 下面Python代码实现输出如下图形，应该在横线处填入的代码是（ ）。

请输入层数: 10

```

1
23
456
7891
23456
789123
4567891
23456789
123456789
1234567891

```

```

1 N = int(input("请输入层数: "))
2
3 K = 1
4
5 for i in range (1, N + 1):
6     for _ in range(1, -----):
7         print(" ", end = "")
8
9     for _ in range(1, -----):
10        print(K, end = "")
11
12        K += 1
13        if K == 10:
14            K = 1
15    print()

```

- ☐ A.

```
1 | N - i + 1
2 | i + 1
```

☐ B.

```
1 | N - i
2 | i
```

☐ C.

```
1 | N
2 | i
```

☐ D.

```
1 | N - i
2 | i + 1
```

## 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** 在集成开发环境里调试程序时，要注意不能修改源程序，因为如果修改，就要终止调试、关闭该文件并重新打开，才能再次开始调试。（ ）

**第 2 题** 在Python代码中，假设N为正整数且大于100，则 `N // 100` 将舍弃个位和十位，如N为1234则 `print(N // 100)` 将输出 12。如果N小于100且大于0，则其值为0。（ ）

**第 3 题** 下列Python代码执行后将输出 `True`，因为 `a` 确实小于 20 和 10。（ ）

```
1 | a = 5
2 | print(a < 10 and 20)
```

**第 4 题** 下面的Python执行后将输出 `True`。（ ）

```
1 | x, y, z = 5, 10, 15
2 | result = x < y < z
3 | print(result)
```

**第 5 题** 下面Python执行后，将输出 10。（ ）

```
1 | a = 10
2 | print(a + True)
```

**第 6 题** 下面Python代码执行时如输入 99.99，将输出 及格 两个汉字。（ ）

```
1 | score = int(input("请输入学生成绩: "))
2 | if score >= 60:
3 |     print("及格")
4 | else:
5 |     print("不及格")
```

**第 7 题** 下面Python代码执行时将导致无限循环（也称死循环）。（ ）

```

1 count = 0
2 while count < 5:
3     count += 1
4     if count == 3:
5         continue
6     print(count, end=' ')

```

第8题 下面的Python代码执行后将输出 `++##2#` 。 ( )

```

1 for i in range(3):
2     if i == 2:
3         continue
4     print("+", end = "#")
5     print(i, end='#')

```

第9题 下列Python代码用于求斐波那契数列，即第1个数0，第2个数1，从第三个数开始，是前两个数之和。如果输入的为大于1的正整数，该代码能输出第N个数的值。 ( )

```

1 n = int(input())
2 a, b = 0, 1
3 for j in range(n):
4     print(a, end=" ")
5     a, b = b, a+b

```

第10题 下面的Python不能实现如下输出，但如果将 `print(0, end='')` 与if对齐，则可以。 ( )

请输入矩阵大小n: 9

```

100000000
020000000
003000000
000400000
000050000
000006000
000000700
000000080
000000009

```

```

1 n = int(input("请输入矩阵大小n: "))
2
3 for i in range(n):
4     for j in range(n):
5         if i == j:
6             print(i + 1, end='')
7             continue
8             print(0, end='')
9     print()

```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：优美的数字
- 时间限制：1.0 s
- 内存限制：512.0 MB

### 3.1.1 题目描述

如果一个正整数的所有数位都相同，小 A 就会觉得这个正整数很优美。例如，正整数 6 的数位都是 6，所以 6 是优美的。正整数 99 的数位都是 9，所以 99 是优美的。正整数 123 的数位不都相同，所以 123 并不优美。

小 A 想知道不超过  $n$  的正整数中有多少优美的数字。你能帮他数一数吗？

### 3.1.2 输入格式

一行，一个正整数  $n$ 。

### 3.1.3 输出格式

一行，一个正整数，表示不超过  $n$  的优美正整数的数量。

### 3.1.4 样例

#### 3.1.4.1 输入样例 1

```
1 | 6
```

#### 3.1.4.2 输出样例 1

```
1 | 6
```

#### 3.1.4.3 输入样例 2

```
1 | 2025
```

#### 3.1.4.4 输出样例 2

```
1 | 28
```

### 3.1.5 数据范围

对于所有测试点，保证  $1 \leq n \leq 2025$ 。

### 3.1.6 参考程序

```
1 # 优美数的数量
2 n = int(input()) # 不大于n的优美数
3 ans = 0 # 保存优美数的数量
4 for i in range(1, n + 1):
5     num = i % 10 # 个位数，优美数肯定与个位数全相同
6     Flag = True # 默认是优美数
7     while i > 0:
8         if num != i % 10:
9             Flag = False # 一旦遇到与个位数不等，不是优美数
10            break # 一旦遇到与个位数不等，终止循环
11        i //= 10
12    if Flag == True: # 如果Flag没有被改变为False，则为优美数
13        ans += 1
14 print(ans)
```



## 3.2 编程题 2

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

### 3.2.1 题目描述

小 A 想绘制一个菱形。具体来说，需要绘制的菱形是一个  $n$  行  $n$  列的字符画， $n$  是一个大于 1 的奇数。菱形的四个顶点依次位于第 1 行、第 1 列、第  $n$  行、第  $n$  列的正中间，使用 # 绘制。相邻顶点之间也用 # 连接。其余位置都是 .。例如，一个 5 行 5 列的菱形字符画是这样的：

```
1 | ..#..
2 | .#.#.
3 | #...#
4 | .#.#.
5 | ..#..
```

给定  $n$ ，请你帮小 A 绘制对应的菱形。

### 3.2.2 输入格式

一行，一个正整数  $n$ 。

### 3.2.3 输出格式

输出共  $n$  行，表示对应的菱形。

### 3.2.4 样例

#### 3.2.4.1 输入样例 1

```
1 | 3
```

#### 3.2.4.2 输出样例 1

```
1 | .#.
2 | #.#
3 | .#.
```

#### 3.2.4.3 输入样例 2

```
1 | 9
```

#### 3.2.4.4 输出样例 2

```
1 | ....#....
2 | ...#.#...
3 | ..#...#..
4 | .#.....#
5 | #.....#
6 | .#.....#
7 | ..#...#..
8 | ...#.#...
9 | ....#....
```

### 3.2.5 数据范围

对于所有测试点，保证  $3 \leq n \leq 29$  并且  $n$  为奇数。

### 3.2.6 参考程序

```
1 # 菱形字符图，n必须为奇数，n是层数
2 n = int(input())
3 k = (n - 1) // 2 # 中间#的位置
4 # 外层i控制层数，内层j控制字符位置
5 for i in range(n):
6     for j in range(n):
7         if j + i == k: # 菱形左上边
8             print('#', end='')
9         elif j - i == k: # 菱形右上边
10            print('#', end='')
11        elif i - j == k: # 菱形左下边
12            print('#', end='')
13        elif i + j == 3 * k: # 菱形右下边
14            print('#', end='')
15        else:
16            print('.', end='')
17    print()
```