



# Python 一级

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

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 小杨这学期刚开学就选修了一门编程课，然后就想编写程序来计算 1 到 10001 之间的所有偶数的和。他希望程序采用简单累加整数的方法，同时希望源程序尽可能清晰、简洁，则下面关于实现累加时采用的主要控制结构的哪个说法最不合适（ ）。

- ☐ A. 使用循环结构
- ☐ B. 使用循环和分支的组合
- ☐ C. 仅使用顺序结构
- ☐ D. 不使用分支结构

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 下列 Python 代码执行后，其输出是（ ）。

```
1 a = 13
2 b = 5
3 print(a / b, a // b, a % b)
```

- ☐ A. 2.6 2 3
- ☐ B. 2.623

☐ C. 2 2 3

☐ D. 2 3 4

第5题 Python表达式  $3 * 3 ** 3 // 3$  的值是( )。

☐ A. 243

☐ B. 81

☐ C. 27

☐ D. 9

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"{N+M}")
```

☐ A. 30

☐ B. 1020

☐ C. {N+M}

☐ D. N+M

第7题 当前是9月，编写Python代码求N个月后的月份。横线处应填入的代码是( )。

```
1 N = int(input())
2 M = _____
3 if M == 0:
4     print(f"{N}个月后是12月")
5 else:
6     print(f"{N}个月后是{M}月")
```

☐ A.  $N \% 12$

☐ B.  $9 + N \% 12$

☐ C.  $(9 + N) // 12$

☐ D.  $(9 + N) \% 12$

第8题 下面Python代码执行后的输出是( )。

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

☐ A. 5050

☐ B. 4950

☐ C. 50

☐ D. 49

第9题 下面的Python代码执行后输出是( )。

```
1 N = 0
2 for i in range(-100,100):
3     N += i % 10
4 print(N)
```

- ☐ A. 90
- ☐ B. 100
- ☐ C. 900
- ☐ D. 1000

第10题 下面Python代码执行后输出是（ ）。

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

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

第11题 下面的Python代码用于求N的镜面数（N的个位到最高位的各位数字依次反过来出现在数字中，但高位0将被忽略，不输出），如输入 1234，则将输出 4321，又如输入 120，则将输出 21，错误的选项是（ ）。

```
1 N = int(input("请输入个位数不为0的正整数: "))
2 rst = 0 #保存逆序结果
3 while -----:
4     rst = rst * 10 + N % 10
5     N = N // 10
6 print(rst)
```

- ☐ A.  $N \neq 0$
- ☐ B.  $\text{not } (N == 0)$
- ☐ C.  $N = 0$
- ☐ D.  $N > 0$

第12题 下面Python代码用于交换两个正整数，其不能实现的代码是（ ）。

☐ A.

```
1 a = int(input("输入第一个正整数: "))
2 b = int(input("输入第二个正整数: "))
3 temp = a
4 a = b
5 b = temp
6 print("a=", a, "b=", b)
```

☐ B.

```
1 a = int(input("输入第一个正整数: "))
2 b = int(input("输入第二个正整数: "))
3 a, b = b, a
4 print("a=", a, "b=", b)
```

☐ C.

```

1 a = int(input("输入第一个正整数: "))
2 b = int(input("输入第二个正整数: "))
3 a = a + b
4 b = a - b
5 a = a - b
6 print("a=", a, "b=", b)

```

☐ D.

```

1 a = int(input("输入第一个正整数: "))
2 b = int(input("输入第二个正整数: "))
3 a = b
4 b = a
5 print("a=", a, "b=", b)

```

第 13 题 有关下面Python代码说法正确的是（ ）。

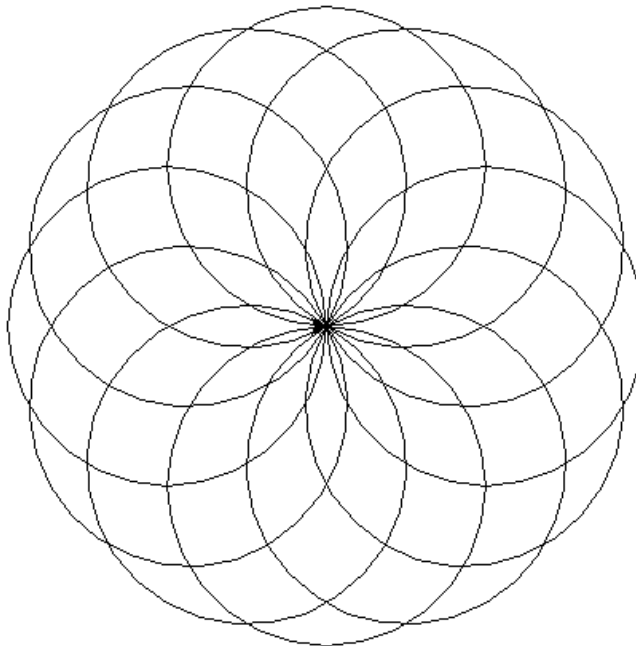
```

1 import turtle
2 turtle.forward(100)

```

- ☐ A. 代码执行后，海龟会水平向右画一条100像素的直线
- ☐ B. 因为在 `turtle.forward(100)` 之前没有执行 `turtle.pendown()`，因此只会水平向右移动100像素，但不会绘制直线
- ☐ C. 因为在 `turtle.forward(100)` 之前没有执行 `turtle.pensize(n)`，其中n为正整数，因此只会水平向右移动100像素，但不会绘制直线
- ☐ D. 因为在 `turtle.forward(100)` 之前没有执行 `turtle.pencolor(color)`，`color` 为合法的颜色值，因此只会水平向右移动100像素，但不会绘制直线或者说绘制了直线，但与背景颜色保持一致，看不出是直线

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



```

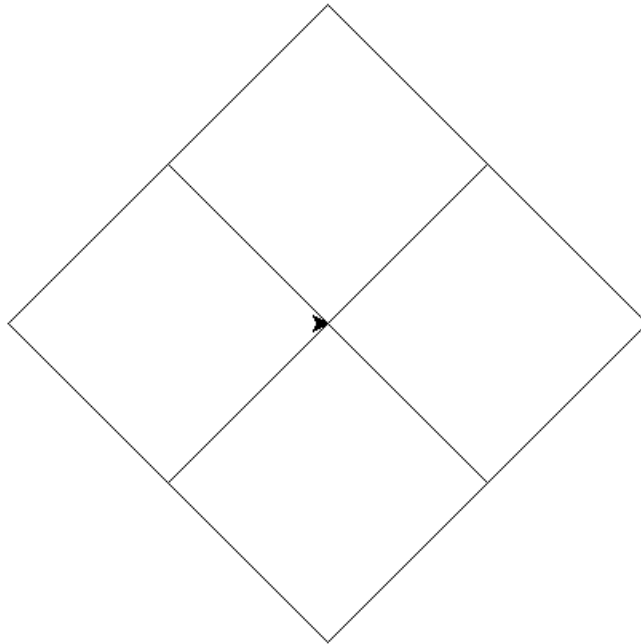
1 import turtle
2 for i in range(12):
3     turtle.circle(100)
4     turtle.right(_____)

```

- ☐ A. 15
- ☐ B. 30

- ☐ C. 45
- ☐ D. 90

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(4):
3     turtle.circle(100, steps = _____)
4     turtle.right(90)
```

- ☐ A. 1
- ☐ B. 2
- ☐ C. 3
- ☐ D. 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 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 执行Python表达式 `10 ** 0.5` 将报错，因为 `0.5` 所在位置只能是正整数。（ ）

第 3 题 下面Python代码执行后将输出 `9`。（ ）

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

第 4 题 下面Python代码执行后将输出 `55`。（ ）

```
1 n = 0
2 for i in range(-10):
3     n = n + i * -1
4 print(n)
```

第5题 将下面Python代码中的 `range(100)` 修改为 `range(1,100)`，其输出与当前代码输出相同。（ ）

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

第6题 将下面Python代码中的 `i < 10` 修改为 `i <= 10`，其执行后输出相同。（ ）

```
1 n, i = 0, 0
2 while i < 10:
3     n += i
4     i += 1
5 print(n)
```

第7题 下面的Python代码执行后将输出 45。（ ）

```
1 n, i = 0, 0
2 while i < 10:
3     i += 1
4     n += i
5 print(n)
```

第8题 执行下面的Python代码后，将输出 30。（ ）

```
1 N, M = 10, 20
2 print(f"{N}+{M}")
```

第9题 下面的Python代码执行后，海龟指向不会发生变化。（ ）

```
1 import turtle
2 for i in range(12):
3     turtle.right(30)
4     continue
```

第10题 下面的Python代码执行后将能绘制出大于3的正多边形。（ ）

```
1 import turtle
2 N = int(input("请输入大于3的正整数: "))
3 for i in range(N):
4     turtle.forward(100)
5     turtle.right(360 % N)
```

### 3 编程题（每题 25 分，共 50 分）

#### 3.1 编程题 1

- 试题名称：商店折扣
- 时间限制：1.0 s
- 内存限制：512.0 MB

### 3.1.1 题目描述

商店正在开展促销活动，给出了两种方案的折扣优惠。第一种方案是购物满  $x$  元减  $y$  元；第二种方案是直接打  $n$  折，也就是说价格变为原先的  $\frac{n}{10}$ 。这里的  $x, y, n$  均是正整数，并且  $1 \leq y < x$ ,  $1 \leq n < 10$ 。

小明在商店挑选了价格总和为  $p$  元的物品，结账时只能使用一种优惠方案。小明最少需要支付多少钱呢？

### 3.1.2 输入格式

四行，四个正整数  $x, y, n, p$ ，含义见题目描述。

### 3.1.3 输出格式

一行，一个小数，表示小明最少需要支付多少钱，保留两位小数。

### 3.1.4 样例

#### 3.1.4.1 输入样例 1

```
1 | 8
2 | 7
3 | 9
4 | 10
```

#### 3.1.4.2 输出样例 1

```
1 | 3.00
```

#### 3.1.4.3 输入样例 2

```
1 | 8
2 | 7
3 | 2
4 | 11
```

#### 3.1.4.4 输出样例 2

```
1 | 2.20
```

### 3.1.5 数据范围

对于所有测试点，保证  $1 \leq y < x \leq 100$ ,  $1 \leq n < 10$ ,  $1 \leq p \leq 100$ 。

### 3.1.6 参考程序

```
1 | # 输入数据
2 | x = int(input()) # 购物x元
3 | y = int(input()) # 购物x元, 返y元
4 | n = int(input()) # 折扣
5 | p = int(input()) # 购买原价p元商品
6 |
7 | # 购物满x元返y元
8 | if p >= x: # 满x元
9 |     a1 = p - y
10 | else: # 不足x元
11 |     a1 = p
12 |
13 | # 按折扣购物
14 | a2 = p * n / 10
15 | # 选择合算方式
16 | if a1 < a2: # 满减合算
```

```
17     print("%.2f" % a1) # print(f"{a1:.2f}")
18 else:
19     print("%.2f" % a2) # print(f"{a2:.2f}")
```

## 3.2 编程题 2

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

### 3.2.1 题目描述

金字塔由  $n$  层石块垒成。从塔底向上, 每层依次需要  $n \times n, (n-1) \times (n-1), \dots, 2 \times 2, 1 \times 1$  块石块。请问搭建金字塔总共需要多少块石块?

### 3.2.2 输入格式

一行, 一个正整数  $n$ , 表示金字塔的层数。

### 3.2.3 输出格式

一行, 一个正整数, 表示搭建金字塔所需的石块数量。

### 3.2.4 样例

#### 3.2.4.1 输入样例 1

```
1 | 2
```

#### 3.2.4.2 输出样例 1

```
1 | 5
```

#### 3.2.4.3 输入样例 2

```
1 | 5
```

#### 3.2.4.4 输出样例 2

```
1 | 55
```

### 3.2.5 数据范围

对于所有测试点, 保证  $1 \leq n \leq 50$ 。

### 3.2.6 参考程序

```
1 # 金字塔石块数量
2 n = int(input()) # 输入金字塔的层数
3 ans = 0 # 答案, 保存石块总数
4 for i in range(1, n + 1):
5     ans += i * i # 累加
6 print(ans) # 输出
```