



Python 二级

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

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 一个无限循环小数是 0.123123123.....，其小数点后第1位是1，后第2位是2，后第3位是3，后第4位是1，依此类推，求第N位。横线处应填入的代码是()。

```
1 N = int(input())
2 remainder = _____
3
4 if remainder == 0:
5     print(1)
6 elif remainder == 1:
7     print(2)
8 else:
9     print(3)
```

- ☐ A. `N % 3`
- ☐ B. `(N - 1) % 3`
- ☐ C. `N // 3`
- ☐ D. `(N - 1) // 3`

第5题 某同学执行Python代码 `print(3 + 3.14)` 时输出 `6.140000000000001`，其原因是()。

- ☐ A. Python中整数和浮点数相加时，整数会被转换为浮点数，而某些十进制小数如 3.14 无法用有限位二进制精确表示，从而产生微小的舍入误差。
- ☐ B. Python的 `print` 函数在输出浮点数时有默认小数点位数，因此输出了更多位数。
- ☐ C. 3.14 是一个无限循环小数，在计算机中无法精确表示。
- ☐ D. 由于计算机CPU的运算错误导致。

第6题 某单位工号的编码规则：编码总长5位，均为数字，前4位依次整除以3，其值累加之和除以10的余数为第5位数字。如某工号为76587，前4位分别整除3后，商分别为2、2、1、2，其累加之和为7，除以10的余数为7，故第5位为7。下面的程序依次输入前4位数字，并计算输出最后一位。程序逐行输入前4位数字，要输出第5位数字。则在横线处应填上的代码是()。

```
1 rst = 0
2 for _ in range(4):
3     N = int(input())
4     rst += _____
5 print(_____)
```

☐ A.

```
1 N % 3
2 rst // 10
```

☐ B.

```
1 N // 3
2 rst % 10
```

☐ C.

```
1 N % 3
2 rst % 10
```

☐ D.

```
1 N // 3
2 rst // 10
```

第7题 下面的Python代码执行之后的输出是()。

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

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

第8题 下面的Python代码执行后其输出是()。

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

- ☐ A. 100
- ☐ B. 55
- ☐ C. 45
- ☐ D. 25

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 == 0:
7             break
8     else:
9         print(i * j, end=" ")
10 else:
11     print(i * j)
```

- ☐ A. 0 0
- ☐ B. 11
- ☐ C. 0
- ☐ D. 0 11

第10题 与下面Python输出效果不一致的代码是()。

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

☐ A.

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

☐ B.

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

☐ C.

```
1 i = 0
2 while True:
3     print(i)
4     i += 1
5     if i >= 10:
6         break
```

☐ D.

```
1 i = 0
2 while True:
3     if i >= 10:
4         break
5     print(i)
6     i += 1
```

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='#')
7 else:
8     print(num)
```

☐ A. 3#6#

☐ B. 3#6#6

☐ C. 1#2#3#4#5#6#

☐ D. 1#2#3#4#5#6#6

第 12 题 下面Python代码执行后，其输出是（ ）。

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

☐ A. 9

☐ B. 10

☐ C. 14

☐ D. 20

第 13 题 漂亮数的定义是：如果N能被M整除，或者某位是M，或者N的每位数之和能被M整除，则说N是M的漂亮数。如果三个条件都满足，则是完整漂亮数。123是3的完整漂亮数，因为123能被3整除，也含有3，其每位数之和为6也能被3整除。下面的代码用于判断N是否为M的完整漂亮数并输出。相关说法正确的是（ ）。

```

1 N = int(input("请输入N, N必须是0或正整数: "))
2 M = int(input("请输入M: M必须大于1小于等于9: "))
3
4 Sum = 0 #记录各位数之和
5
6 Flag = False #记录N是否含有M
7
8 while N != 0:
9     num = N % 10
10
11     Sum += num
12     if num == M:
13         Flag = True
14
15     N //= 10
16
17 if (N % M == 0) and (Flag == True) and (Sum % M == 0):
18     print(f"{N}是{M}的完整漂亮数")
19 else:
20     print(f"{N}不是{M}的完整漂亮数")

```

- ☐ A. 代码能完成题目设定目标。
- ☐ B. 在 while 循环中 N 最终将变成0, 因为代码 `N % M` 将总是满足条件, 可以在 while 前增加一行代码 `old_num = N`, 并将 `N % M` 改为 `old_num % M`, 随后 `print()` 中的 `{N}` 改为 `{old_num}`。
- ☐ C. while 循环中 if 判断语句可以增加 else 子句, 其内容为 `Flag = False`。
- ☐ D. 如果先后输入 0 和 3, 则肯定会输出 0是3的完整漂亮数。

第 14 题 阅读下面的Python代码。执行后如输入 5, 其输出的字符图形是 ()。

```

1 n = int(input())
2 for i in range(n):
3     for j in range(n - i - 1):
4         print(" ", end="")
5     for k in range(2 * i + 1):
6         print("*", end="")
7     print()

```

☐ A.

```

1 *****
2 *****
3 ****
4 ***
5 **
6 *

```

☐ B.

```

1 *
2 ***
3 *****
4 *****
5 *****

```

☐ C.

```

1 *
2 **
3 ***
4 ****
5 *****

```

☐ D.

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

第 15 题 某学校举办“十佳歌手大奖赛”，经过选拔最终参赛选手有25人，评委10人，最终计分规则去掉一个最高分去掉一个最低分作为该参赛选手的最终得分，并输出该得分。每位评委都必须打分，且分数都大于等0。相关说法正确的是（ ）。

```
1 for _ in range(25):
2
3     max_score = 0 #记录最高分
4     min_score = 9999 #记录最低分
5
6     total_score = 0 #记录总分
7
8     for _ in range(10):
9
10        now_score = float(input()) #录入评委打分
11
12        if now_score > max_score:
13            max_score = now_score
14
15        if now_score < min_score:
16            min_score = now_score
17
18        total_score += now_score
19
20    print(total_score - max_score - min_score)
```

- ☐ A. 程序总体逻辑错误。因为要去掉最高分和最低分，需要排序，而程序没有相关代码。
- ☐ B. 内层循环和外层循环之间的三行代码也就是 `max_score = 0` 开始的三行代码应该移动到外层循环外。
- ☐ C. `if now_score > max_score` 和 `if now_score < min_score` 中的 `>` 和 `<` 分别调整为 `>=` 和 `<=` 不影响程序执行结果。
- ☐ D. `total_score += now_score` 不可以更改为 `total_score = total_score + now_score`。

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 `5 < 10 and 20` 的值为 `True`。（ ）

第 3 题 Python表达式 `10 ** 0.5 == 10 ** 1 / 2` 的值为 `False`。（ ）

第 4 题 下面Python代码执行时如果输入整数（包括负整数、0 和正整数），其输出都将是 0。（ ）

```
1 N = int(input())
2 while N:
3     N //= 10
4 print(N)
```

第5题 下面的Python代码执行后，其输出是 4 False 。（ ）

```
1 a = 4
2 b = a == 5
3 print(a, b)
```

第6题 下面Python代码执行后将输出 5 。（ ）

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

第7题 下面的Python代码可以用于判断正整数N的位数（即几位数，如 123 是 3 位数， 12 为 2 位数）。（ ）

```
1 N = int(input())
2 i = 1
3 while True:
4     if N % (10 ** i) == N:
5         print(f"{N}是{i}位数")
6         break
7     i += 1
```

第8题 计算交叉加减的结果，形如 1-2+3-4+5-.....。将下面Python代码中的 Flag = -Flag 改为 Flag -= Flag 效果相同。（ ）

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

第9题 下列Python代码执行后将输出 55 。（ ）

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

第10题 下面Python代码执行后输出如下，因为代码 print() 的括号中没有任何内容，删除不影响输出效果。（ ）

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

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

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称：环保能量球
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.1.1 题目描述

小杨最近在玩一个环保主题的游戏。在游戏中，小杨每行走 1 公里就可以获得 1 点“环保能量”。

为了激励玩家，游戏设置了“里程奖励”：小杨每行走 x 公里，游戏就会额外奖励 1 点能量。

现在已知小杨总共行走了 n 公里，请你帮他计算，他一共能获得多少点环保能量？

3.1.2 输入格式

第一行包含一个正整数 t ，代表测试数据组数。

对于每组测试数据：

- 第一行包含一个正整数 n ，代表行走的公里数。
- 第二行包含一个正整数 x ，代表奖励触发的间隔。

3.1.3 输出格式

对于每组测试数据，输出一个整数，代表小杨获得的环保能量总数。

3.1.4 样例

3.1.4.1 输入样例

```
1 | 3
2 | 5
3 | 2
4 | 10
5 | 3
6 | 2
7 | 5
```

3.1.4.2 输出样例

```
1 | 7
2 | 13
3 | 2
```

3.1.4.3 样例解释

- 对于第 1 组数据， $n = 5, x = 2$ ：小杨行走获得 5 点能量。此外，他在第 2 公里和第 4 公里时各获得 1 点额外奖励，总共 $5 + 2 = 7$ 点。
- 对于第 2 组数据， $n = 10, x = 3$ ：行走获得 10 点。他在第 3、6、9 公里时各获得 1 点额外奖励，总共 $10 + 3 = 13$ 点。
- 对于第 3 组数据， $n = 2, x = 5$ ：行走获得 2 点。由于行走路程不足 5 公里，没有额外奖励，总共 2 点。

3.1.5 数据范围

对于全部数据，保证： $1 \leq t \leq 100$ ， $1 \leq n, x \leq 1000$ 。

3.1.6 参考程序

```
1 # 读取测试数据组数
2 line_count = int(input()) # 读取数据组数
3
4 for _ in range(line_count):
5     # 一行读取一个数
6     n = int(input()) # 读取行走的公里数
7     x = int(input()) # 读取奖励间隔数
8
9     ans = n # 基础奖励点数
10    add_point = n // x
11    print(ans + add_point) #输出基础奖励点数和额外奖励数
```

3.2 编程题 2

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

3.2.1 题目描述

小杨在探险时发现了一张神奇的矩形地图，地图有 H 行和 W 列。每个格子的坐标是 (r, c) ，其中 r 表示行号从 1 到 H ， c 表示列号 1 到 W 。

小杨听说地图中隐藏着一些“黄金格”，这些格子满足一个神秘的数学挑战：当格子坐标 (r, c) 代入特定的不等式关系成立时，该格子就是黄金格。具体来说，黄金格的条件是： $\sqrt{r^2 + c^2} \leq x + r - c$ 。

例如，如果参数 $x = 5$ ，那么格子 $(4, 3)$ 就是黄金格。因为左边坐标平方和的平方根 $\sqrt{4^2 + 3^2}$ 算出来是 5，而右边 $5 + 4 - 3$ 算出来是 6，5 小于等于 6，符合条件。

3.2.2 输入格式

三行，每行一个正整数，分别表示 H, W, x 。含义如题面所示。

3.2.3 输出格式

一行一个整数，代表黄金格数量。

3.2.4 样例

3.2.4.1 输入样例 1

```
1 | 4
2 | 4
3 | 2
```

3.2.4.2 输出样例 1

```
1 | 4
```

3.2.4.3 样例解释

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

图中标注为黄色的四个格子是黄金格，坐标分别为 (1,1)，(2,1)，(3,1)，(4,1)。

3.2.5 数据范围

对于所有测试点，保证给出的正整数不超过 1000。

3.2.6 参考程序

```
1 H = int(input()) #矩形高度, 编号1-H
2 W = int(input()) #矩形宽度, 编号1-W
3 x = int(input()) #特别常数
4
5 ans = 0
6
7 #遍历每一个格子, 求出(r^2 + c^2)的平方根, 与x + r - c进行比较
8 for r in range(1, H + 1):
9     for c in range(1, W + 1):
10         if ((r * r + c * c) ** 0.5) <= x + r - c:
11             ans += 1
12 print(ans)
```