



Python 二级

2026 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	C	B	A	D	A	C	B	D	B	D	A	C	A	A	A

第 1 题 第二届世界人形机器人运动会开幕式上，机器人方阵通过控制程序来完成整场表演。表演前，工程师们使用集成开发环境（IDE）编写并调试方阵的控制程序。下列关于 IDE 功能的描述，错误的是（ ）。

- ☐ A. 可以提供代码自动补全，提高编程效率
- ☐ B. 可以提供辅助信息，帮助排查程序逻辑错误
- ☐ C. 可以直接控制机器人电机转动
- ☐ D. 支持语法高亮，使代码更易阅读

第 2 题 计算机 CPU 中的高速缓冲存储器（Cache）主要为提升系统性能而设计。下列关于 Cache 的说法中正确的是（ ）。

- ☐ A. Cache 的容量比主存大，速度比主存快
- ☐ B. Cache 通常用于暂存 CPU 频繁访问的数据
- ☐ C. Cache 中的数据断电后不会丢失，属于非易失性存储器
- ☐ D. Cache 的访问速度比 CPU 寄存器还要快

第 3 题 下列 Python 代码执行后，其输出是（ ）。

```
1 | print(3 == 3.0)
```

- ☐ A. True
- ☐ B. true
- ☐ C. False
- ☐ D. false

第 4 题 下列 Python 代码执行后，其输出是（ ）。

```
1 | print(bool(5) + bool(-5) + bool(5.0))
```

- ☐ A. True
- ☐ B. False
- ☐ C. 2
- ☐ D. 3

第5题 判断年份 `N` 是否为闰年的规则是：能被4整除但不能被100整除，或者能被400整除。以下Python表达式中，能正确判断 `N` 是否为闰年的是（ ）。

- ☐ A. `(N % 4 == 0 and N % 100 != 0) or (N % 400 == 0)`
- ☐ B. `N % 4 == 0 and N % 100 == 0 or N % 400 == 0`
- ☐ C. `(N / 4 == 0 and N / 100 != 0) or (N / 400 == 0)`
- ☐ D. `(N % 4 == 0 and N % 100 != 0) and (N % 400 == 0)`

第6题 下列Python代码执行后，其输出是（ ）。

```
1 tot = 0
2 for i in range(10):
3     if i:
4         tot += 1
5 print(tot)
```

- ☐ A. 0
- ☐ B. 8
- ☐ C. 9
- ☐ D. 10

第7题 执行如下Python代码，当输入4时，其输出是（ ）。

```
1 n = int(input())
2 tot = 0
3 for i in range(1,n):
4     for j in range(1,i):
5         tot += 1
6 print(tot)
```

- ☐ A. 1
- ☐ B. 3
- ☐ C. 6
- ☐ D. 10

第8题 下列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 += i
5 print(tnt)
```

- ☐ A. 12
- ☐ B. 18
- ☐ C. 22
- ☐ D. 27

第9题 下列Python代码执行后，其输出是（ ）。

```

1 tot = 0
2 for i in range(-5, 5):
3     if not (i % 3 != 0):
4         tot += 1
5 print(tot)

```

- ☐ A. 2
- ☐ B. 3
- ☐ C. 7
- ☐ D. 9

第 10 题 有关如下 Python 代码描述，错误的是（ ）。

```

1 n = int(input("请输入一个正整数: "))
2 if n <= 1:
3     print(f"{n}非素数")
4 elif n == 2:
5     print("2是素数")
6 else:
7     limit = int(n ** 0.5) + 1
8     for i in range(2, limit):
9         if n % i == 0:
10            print(f"{n}非素数")
11            break
12 else:
13     print(f"{n}是素数")

```

- ☐ A. `n <= 1` 条件用于判断输入 0、1 以及负数时的情况，不能省略。
- ☐ B. `n == 2` 条件及其相关代码可以省略，能正确输出 2是素数。
- ☐ C. `break` 语句不可以省略，否则非素数还将执行最后一行的语句。
- ☐ D. `limit = int(n ** 0.5) + 1` 语句中的 1 可以修改为其他大于等于 1 的正整数，执行结果不会有错误，但循环次数增多，效率有所影响。

第 11 题 下列 Python 代码执行后，其最终输出结果是（ ）。

```

1 total = 0
2 i = 1
3 while i <= 3:
4     for j in range(1, i + 1):
5         total += i * j
6         i += 1
7 print(total)

```

- ☐ A. 9
- ☐ B. 11
- ☐ C. 23
- ☐ D. 25

第 12 题 下列 Python 代码执行后，其输出是（ ）。

```

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

```

- ☐ A. 2 3 2
- ☐ B. 3 0 2
- ☐ C. 3 1 2
- ☐ D. 3 2 3

第 13 题 如下的 Python 代码用于输出如下数字图形。注意：最后一行数字之前没有空格。横线处应填入的代码是（ ）。

```

1 请输入行数: 12
2      0
3      12
4      345
5      6789
6      01234
7      567890
8      1234567
9      89012345
10     678901234
11     5678901234
12     56789012345
13     678901234567

```

```

1 i = 0
2 line = int(input("请输入行数: "))
3 for j in range(line):
4     for _ in range(_____):
5         print(" ", end = "")
6     for _ in range(j + 1):
7         print(i, end = "")
8         i += 1
9         if i >= 10:
10            i = 0
11 print()

```

- ☐ A. line - j - 1
- ☐ B. line - j
- ☐ C. line
- ☐ D. line + 1

第 14 题 小北的身高是 1.65 米。班上有 `n` 个小朋友，小北想和身高最接近自己的人交朋友。如果有多小朋友与小北的身高差相同，小北想和其中身高较矮的第一个小朋友做朋友。横线处应该填入的代码是（ ）。

```

1 n = int(input("请输入小朋友的人数: "))
2 target = 1.65
3 best_height = 0.0
4 min_diff = 100.0
5
6 for i in range(n):
7     h = float(input(f"请输入第 {i} 个小朋友的身高: "))
8     diff = abs(h - target)
9
10    if _____: # 横线处填空
11        min_diff = diff
12        best_height = h
13
14 print("小北的朋友身高为:", best_height)

```

- ☐ A. `diff < min_diff or (diff == min_diff and h < best_height)`
- ☐ B. `diff <= min_diff and h < best_height`
- ☐ C. `diff < min_diff and h < best_height`
- ☐ D. `diff < min_diff or h < best_height`

第15题 小北的身高是1.65米。班上有 n 个小朋友，小北想和身高最接近自己的人交朋友。如果有多个小朋友与小北的身高差相同，小北想和其中身高较矮的第一个小朋友做朋友。横线处应该填入的代码是（ ）。

```

1 n = int(input("请输入小朋友的人数: "))
2 target = 1.65
3 best_height = 0.0
4 min_diff = 100.0 # 假设身高差不会超过100
5
6 for i in range(n):
7     h = float(input(f"请输入第 {i + 1} 个小朋友的身高: "))
8
9     if h > target:
10        diff = h - target
11    else:
12        diff = target - h
13
14    if diff < min_diff:
15        min_diff = diff
16        best_height = h
17    elif diff == min_diff:
18        _____
19        _____
20
21 print("小北的朋友身高为:", best_height)

```

☐ A.

```

1 if h < best_height:
2     best_height = h

```

☐ B.

```

1 if h > best_height:
2     best_height = h

```

☐ C.

```

1 if h < target:
2     best_height = h

```

☐ D.

```
1 | if h <= target:
2 |     best_height = h
```

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 Python 程序执行 `input()` 函数时通常是使用键盘等输入设备来输入某个变量的值。

第 2 题 执行 Python 语句 `print( 5 and 2 )` 将输出 `True` 。

第 3 题 Python 表达式 `2 * - 2 ** 4` 的值是 `32` 。

第 4 题 下列 Python 代码执行后，其输出是 `True` 。

```
1 | a = 5
2 | a == 5
3 | print(a)
```

第 5 题 执行 Python 代码 `print(not not 5)` 将输出 `5` 。

第 6 题 下列 Python 代码执行后，其输出是 `369` 。

```
1 | for i in range(1,10):
2 |     if not i % 3:
3 |         print(i, end = "")
```

第 7 题 下列 Python 代码执行后，其输出是 0 到 `N` 之间（包含 `N`）所有整数之和。不限定 `N` 为正整数。

```
1 | N = int(input())
2 | tot = 0
3 | for i in range(N + 1):
4 |     tot += i
5 | print(tot)
```

第 8 题 下列 Python 代码执行后，其输出是 `0` 。

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

第 9 题 求解鸡兔同笼。在如下 Python 代码中，`tou` 代表鸡和兔的头总数，`jiao` 代表鸡和兔的脚总数。先后两个输入均为正整数。该程序能求解鸡兔同笼。如无解则输出 `无解` 两个汉字。

```

1  tou = int(input("请输入鸡和兔的头总数: "))
2  jiao = int(input("请输入鸡和兔的脚总数: "))
3
4  for tu_tou in range(tou + 1):
5      ji_tou = tou - tu_tou
6      if tu_tou * 4 + ji_tou * 2 == jiao:
7          print(f"兔的数量={tu_tou},鸡的数量={ji_tou}")
8  if tu_tou == tou:
9      print("无解")

```

第10题 在下列 Python 代码中，如果 N 和 M 均输入大于 0 的整数，则可以取得正整数 N 的从右到左第 M 位置的数。个位数为第 1 位置， M 在 N 的位数范围内。

```

1  N = int(input("请输入正整数:"))
2  M = int(input("从右到左第几位? "))
3
4  print(N // (10 ** (M - 1)) % 10)

```

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称：直角三角形
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.1.1 题目描述

小双刚学会勾股定理，得知直角三角形的斜边的平方等于两直角边的平方和。

小双创造了直角三角形数列：数列前两项由小双指定，从第三项开始，数列中的每个数字的平方等于前两项的平方和。

例如，第一项是 3.0，第二项是 4.0，则第三项是 $\sqrt{3.0^2 + 4.0^2} = 5.0$ 。

其中 $\sqrt{x}$ 表示 x 的算术平方根，可以使用 C/C++/Python 标准库中对应函数计算。

小双给定了数列的前两项，并给定一个数列上限，想让你算算数列的第几项会超过小双给定的数列上限？

3.1.2 输入格式

输入三行，第一行一个浮点数是数列第一项，第二行一个浮点数是数列第二项，第三行一个浮点数表示数列上限。

3.1.3 输出格式

输出一个正整数，表示第几项开始会超过小双给定的数列上限。

3.1.4 样例

3.1.5 输入样例 1

```

1  3.0
2  4.0
3  10.0

```

3.1.6 输出样例 1

```

1  6

```

3.1.7 样例解释 1

数列依次为：

3.0,

4.0,

$$\sqrt{3.0^2 + 4.0^2} = 5.0,$$

$$\sqrt{4.0^2 + 5.0^2} = 6.403124 \dots,$$

$$\sqrt{5.0^2 + 6.403124^2} = 8.124038 \dots,$$

$$\sqrt{6.403124^2 + 8.124038^2} = 10.344080 \dots,$$

因此第 6 项会超过上限。

3.1.8 数据范围

输入均不超过 10000，且两个初始项与数列上限均为正数；若上限不大于初始项，输出 3。

题目保证给定第一项小于第二项。

3.1.9 参考程序

```
1  # 引入math的sqrt函数，进行开根运算
2  from math import sqrt
3  # 记录a,b,U
4  a = float(input())
5  b = float(input())
6  U = float(input())
7  ans = 1
8  if a <= U:
9      ans += 1
10 # 根据最初的a,b，不断计算下一项，判断其是否小于等于U
11 while b <= U:
12     ans += 1
13     # 也可以使用(a**2+b**2)**0.5来得到下一条边的长度
14     a, b = b, sqrt(a**2 + b**2)
15 # 最终的答案
16 print(ans)
```

3.2 编程题 2

- 试题名称：字符变换
- 时间限制：1.0 s
- 内存限制：512.0 MB

3.2.1 题目描述

小明有一个 8×16 的画布（8 行 16 列），画布中的格子从上到下从左到右编号，左上的编号为 1，第一排最后一个的编号为 16，最后一排第一个编号为 113，右下为 128。

初始画布中的每个格子都是 *，随后小明会给定三个正整数，小明会将对应编号的格子变为该正整数对应的 ASCII 编码字符。

最后请帮小明输出整个画布。

3.2.2 输入格式

输入包含三行，每行一个正整数，表示需要替换的画布格子编号。

3.2.3 输出格式

输出 8×16 的画布。

3.2.4 样例

3.2.5 输入样例 1

```
1 48
2 65
3 97
```

3.2.6 输出样例 1

```
1 *****
2 *****
3 *****0
4 *****
5 A*****
6 *****
7 a*****
8 *****
```

3.2.7 数据范围

输入范围是 48 到 57，65 到 90，以及 97 到 122（即 ASCII 编码的 0 到 9，A 到 Z，以及 a 到 z）。

3.2.8 参考程序

```
1 # 记录a,b,c
2 a = int(input())
3 b = int(input())
4 c = int(input())
5 p = 0
6 # 挨个位置输出字符，判断其编号是否与a,b,c相同
7 for i in range(8):
8     for j in range(16):
9         p += 1
10        # 如果当前的位置是a,b,c中的一个，那么就打印a,b,c对应的符号
11        if p == a:
12            print(chr(a), end='')
13        elif p == b:
14            print(chr(b), end='')
15        elif p == c:
16            print(chr(c), end='')
17        else:
18            print('*', end='')
19    # 每一行结束换行
20    print()
```