CCF 编程能力等级认证
Grade Examination of Software Programming

Python 二级

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

第 1 题 2026年春节联欢晚会上一个武术表演节目《武 BOT》。节目中多个人形机器人会表演空翻，它们落地可能会有微微踉跄，但都会迅速调整姿态站稳，并适当移动来和前后左右的其他机器人保持原来队列。如果将机器人视作一个计算机系统，那么在该计算机系统中下面哪一项不能作为输入设备( )。

- A. 用于检测重心的重力传感器
- B. 预装的AI算法程序
- C. 接收动作指令的遥控器
- D. 拍摄其他机器人的摄像头

第 2 题 下面代码用来找出输入的 N 个正整数中最大的一个。如果将代码段用流程图来表示，则 `L1` 标记的代码行应该使用的图形是（ ）。

```
Max = 0
N = int(input())

while(N):
    val = int(input())
    if val > Max: #L1
        Max = val
    N -= 1

print(Max)
```

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

第 3 题 有关下面 Python 的说法，正确的是( )。

```
PI = 3.1415926
print(PI)
```

- A. 为了方便初学者， `print(PI)` 和 `print(pi)` 效果相同，即变量的大小写不敏感
- B. `print(PI)` 修改为 `print(Pi)` 能正常执行
- C. 不能用PI做变量名，因为要保存圆周率这个常量

- [ ] **D.** 将程序中 `PI` 全部改写为 `Pai`，将能正常执行，不会报错

第 4 题 下面选择项中，与 Python 表达式 `not (x > 5 and y <= 10)` 等价的是(　)。

- [ ] **A.** `x <= 5 and y > 10`
- [ ] **B.** `x > 5 or y <= 10`
- [ ] **C.** `x <= 5 or y > 10`
- [ ] **D.** `not x > 5 and not y <= 10`

第 5 题 某同学执行 Python 代码 `print((0.1 + 0.2) == 0.3)` 时输出 `False`，其原因是(　)。

- [ ] **A.** Python 的 `+` 运算符在处理小数时存在 bug
- [ ] **B.** 0.1、0.2 和 0.3 在计算机中无法用二进制浮点数精确表示，导致 0.1 + 0.2 的结果与 0.3 存在微小误差
- [ ] **C.** `==` 运算符不能用于比较浮点数，只能用于整数
- [ ] **D.** 因为 0.1 + 0.2 的数学结果不等于 0.3

第 6 题 下面的 Python 代码执行后其输出是(　)。

```python
tnt = 0
for i in range(5):
    for j in range(i):
        tnt += 1
    print(tnt, end = "#")
print(tnt)
```

- [ ] **A.** `0#1#3#6#10#10`
- [ ] **B.** `1#2#3#4#5#6#7#8#9#10#10`
- [ ] **C.** `10#10`
- [ ] **D.** `10`

第 7 题 下面的 Python 代码执行之后的输出是(　　)。

```python
for i in range(-2, 2):
    if not i % 3:
        print(i, end = "#")
```

- [ ] **A.** `0#`
- [ ] **B.** `-2#-1#1#`
- [ ] **C.** `-1#0#`
- [ ] **D.** `-2#0#1#`

第 8 题 下面的 Python 代码执行后其输出是(　)。

```python
cnt = 0
for i in range(1, 5):
    for j in range(i):
       print(j,end = "#")
    break
else:
    print(i*j)
```

- [ ] **A.** `0#0#1#0#1#2#0#1#2#3#12`

- [ ] **B.** `0#0#1#0#1#2#0#1#2#3#`
- [ ] **C.** `0#`
- [ ] **D.** `1#`

第 9 题 下面 Python 代码执行后其输出是(　　)。

```
count = 0
for i in range(1, 4):
    for j in range(1, 5):
        if j == 3:
            continue
        if i == 2:
            break
        count += 1
print(count)
```

- [ ] **A.** `2`
- [ ] **B.** `4`
- [ ] **C.** `6`
- [ ] **D.** `8`

第 10 题 下面4个选项中，与下面 Python 代码输出结果相同的是(　　)。

```
i = 0
while i < 5:
    print(i)
    i += 1
```

- [ ] **A.**

```
for i in range(5):
    print(i)
```

- [ ] **B.**

```
for i in range(1, 5):
    print(i)
```

- [ ] **C.**

```
for i in range(6):
    print(i)
```

- [ ] **D.**

```
for i in range(0, 6):
    print(i)
```

第 11 题 下面 Python 代码执行后输出是（　）。

```
n = 10
while n > 0:
    n -= 1
    if n % 3 == 0:
        continue
    if n == 5:
        break
print(n)
```

- [ ] **A.** `0`
- [ ] **B.** `5`
- [ ] **C.** `6`
- [ ] **D.** `7`

第 12 题 下面 Python 代码执行后，其输出是（ ）。

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

- [ ] **A.** `5`
- [ ] **B.** `15`
- [ ] **C.** `20`
- [ ] **D.** `30`

第 13 题 某学校图书馆的借阅卡号由6位数字组成。前5位是顺序编号，第6位是校验码，用于防止输错。校验码规则如下：将前5位数字相加，然后除以10的余数，就是第6位数字。如卡号123455的前5位之和为15，除以10的余数是5，故第6位为5。下面的 Python 代码用于判断卡号是否正确，横线处应填入的代码是(　)。

```
N = int(input("请输入卡号: "))
order_num = N // 10 #获得前5位顺序号,假设录入一定为6位正整数
check_num = N % 10 #获得最后一位

tnt = 0 #保存前5位之和
for i in range(5):
    ________________
    order_num //= 10

if ___________________:
    print("符合校验规则")
else:
    print("不符合校验规则")
```

- [ ] **A.**

```
tnt += order_num // 10
tnt // 10 == check_num
```

- [ ] **B.**

```
tnt += order_num % 10
tnt % 10 == check_num
```

- [ ] **C.**

```
tnt = order_num % 10 + tnt
tnt % 10 = check_num
```

- [ ] **D.**

```
tnt = order_num % 10
tnt // 10 == check_num
```

第 14 题 下面的 Python 代码执行后其输出的数字图形是（ ）。

```
for i in range(1, 5):
    for j in range(1, i + 1):
        print(j, end='')
    print()
```

☐ **A.**

```
1
12
123
1234
```

☐ **B.**

```
1
22
333
4444
```

☐ **C.**

```
1
21
321
4321
```

☐ **D.**

```
4
34
234
1234
```

第 15 题 某学校举办“校园演讲比赛”，每位选手由8位评委打分（分数为 0~10 的整数），且每位评委必须打分。计分规则：若至少有5位评委给出大于等于6分，则成绩有效，最终得分为所有8位评委的总分；否则记为0分。以下程序通过键盘依次输入8个分数，并计算最终得分。横线处应填入（ ）。

```
total_score = 0 # 所有分数之和
high_count = 0 # ≥6分的评委数量

for i in range(8):
    score = int(input("请输入评委分数: "))
    --------------------
    if score >= 6:
        ----------------

if high_count >= 5:
    print(total_score)
else:
    print(0)
```

☐ **A.**

```
total_score += score
high_count += 1
```

☐ **B.**

```
total_score += score
high_count += score
```

C.

```
high_count += 1
total_score += score
```

D.

```
total_score *= score
high_count *= 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 小明的妈妈最近刚刚给他买了一块电话手表，除了可以看时间，小明也可以用它和妈妈打电话、收发信息，那么可以推测这块手表中装有一款特定操作系统。（ ）

第 2 题 Python 代码 `print(4 ** 2 == 2 * 2 ** 2)` 执行后的输出是 `True`。（ ）

第 3 题 下面的 Python 代码执行时将报错，因为10是整数类型，`a` 是布尔类型。由于类型不同，不能进行加法运算。（ ）

```
a = True
print(10 + a)
```

第 4 题 下面 Python 代码执行后将输出 `0-3-6-9-`。（ ）

```
for i in range(10):
    if i % 3:
        continue
    print(i, end = "-")
```

第 5 题 执行下面的 Python 代码，其语句 `print(N)` 将被执行 0 次或极多次（即死循环）。（ ）

```
N = input()
while N:
    print(N)
```

第 6 题 在下面的 Python 代码中，删除 `continue` 不影响执行效果。（ ）

```
for  i in range(10):
    i += 1
    continue
print(i)
```

第 7 题 下列 Python 代码用于计算从 1 到 N（含 N）的所有正整数中，数字 `3` 出现的总次数。当 `N = 40` 时，共有14个3，包括3、13、23、30-39，其中33计两次，共计14个3。将代码 `while i != 0` 改为 `while i` 执行结果相同。（ ）

```python
N = int(input("请输入正整数N: "))
cnt = 0 #保存3的个数
for  i in range(1, N + 1):
    while i != 0:
        if i % 10 == 3:
            cnt += 1
        i //= 10
print(cnt)
```

第 8 题 下面的 Python 代码执行将不会有输出，因为内层循环 `j` 总是 0 开始，`i * j % 10 == 0` 将会被满足，执行 `break`，故而不会执行代码中的 `else` 子句部分。（ ）

```python
for i in range(1, 10):
    for j in range(i):
        if i * j % 10 == 0:
            break
else:
    print(i*j)
```

第 9 题 下列 Python 代码执行后将输出 `1#4#9#16#16`。（ ）

```python
for  i in range(1, 5):
    for j in range(1, i + 1):
        if i * j % 10 == 0:
            break
    else:
        print(i * j, end = "#")
else:
    print(i * j)
```

第 10 题 下面 Python 代码执行后输出如左图所示，将 `f" {i*j}"` 修改为 `f"{i*j:3}"` 即可实现右图输出。（ ）

```
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  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
```

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

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

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

3.1.1 题目描述

对于正整数 n，如果 n 的所有数位中包含**恰好** 3 个 2，Alice 会认为这个正整数是美丽的。例如，正整数 24122 中包含 3 个 2，所以 24122 **是**美丽的；正整数 132 中包含 1 个 2，所以 132 **不是**美丽的；正整数 212322 中包含 4 个 2，所以 212322 **不是**美丽的。

Alice 想知道正整数 L 到正整数 R （包括 L 和 R）中有多少美丽的正整数，你能帮她数一数吗?

3.1.2 输入格式

输入共 2 行，第一行为正整数 L，第二行为正整数 R。

3.1.3 输出格式

输出一行，包含一个整数 C，表示 L 到 R 中有 C 个美丽数。

3.1.4 样例

3.1.4.1 输入样例

```
2221
2223
```

3.1.4.2 输出样例

```
2
```

3.1.5 样例解释

2221 到 2223 中，2221 与 2223 是美丽的，2222 不是美丽的。

3.1.6 数据范围

保证 $1 \leq L \leq R \leq 10^6$。

3.1.7 参考程序

```python
left = int(input()) # 范围开始
right = int(input()) # 范围结束

total_count = 0 # 美丽数的总数量

for i in range(left, right + 1):

    count2 = 0 # 记录i中2的数量，每次循环清零

    while i > 0:
        if i % 10 == 2: # 得到个位数，并判断是否为2
            count2 += 1 # 如果为2，count2累加1

        i //= 10 # 去除个位数后的新数

    if count2 == 3: # 判断2的数量是否为3
        total_count += 1 #2的数量为3，美丽数，累加1

print(total_count) # 输出美丽数的数量
```

3.2 编程题 2

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

3.2.1 题目描述

输入一个正整数 n，你需要绘制一个 n 行 n 列的正方形，绘制规则如下：

- 正方形的四个顶点使用 `+` 绘制；
- 除顶点外，第 1 行与第 n 行使用 `-` 绘制；
- 除顶点外，第 1 列与第 n 列使用 `|` 绘制；
- 正方形内部使用 `*` 绘制。

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，表示对应的正方形。

3.2.4 样例

3.2.4.1 输入样例

```
5
```

3.2.4.2 输出样例

```
+---+
|***|
|***|
|***|
+---+
```

3.2.5 数据范围

保证 $3 \leq n \leq 100$。

3.2.6 参考程序

```python
n = int(input()) #输入n行n列的n值

for i in range(0, n):# 一共n行，循环n次
    for j in range(0, n): # 一共n列，循环n次

        if j == 0 or j == n - 1: # j == 0为第1列，j == n - 1为最后n列

            if i == 0 or i == n - 1: # 第一行或者最后一行的处理
                print('+', end='')
            else:
                print('|', end='') # 非第一行或者最后一行的处理

        else: # 不是第1列或者最后一列

            if i == 0 or i == n - 1: # 第1行或者最后一行的处理
                print('-', end='')
            else:
                print('*', end='')# 非第1行或者最后一行的处理

    print() #一行结束，换行
```