

## 2023 年 6 月 GESP Python 一级试卷解析

CCF 编程能力等级认证，英文名 Grade Examination of Software Programming（以下简称 GESP），由中国计算机学会发起并主办，是为青少年计算机和编程学习者提供学业能力验证的平台。GESP 覆盖中小学全学段，符合条件的青少年均可参加认证。GESP 旨在提升青少年计算机和编程教育水平，推广和普及青少年计算机和编程教育。

GESP 考察语言为图形化（Scratch）编程、Python 编程及 C++编程，主要考察学生掌握相关编程知识和操作能力，熟悉编程各项基础知识和理论框架，通过设定不同等级的考试目标，让学生具备编程从简单的程序到复杂程序设计的编程能力，为后期专业化编程学习打下良好基础。

本次为大家带来的是 2023 年 3 月份 Python 一级认证试卷真题解析。

### 一、单选题（每题 2 分，共 30 分）

1. 以下不属于计算机输出设备的有（ ）。

- A. 麦克风
- B. 音箱
- C. 打印机
- D. 显示器

【答案】A

【解析】考察计算机基础知识，输出指从计算机内部向外，输入指从外部向内，本题中 B、C、D 三个选项均是从内部向外，只有 A 是接收声音并传入计算机，是从外部向内。

2. ChatGPT 是 OpenAI 研发的聊天机器人程序，它通过理解和学习人类的语言来进行对话，还能根据聊天的上下文进行互动，完成很多工作。请你猜猜看，下面任务中，ChatGPT 不能完成的是（ ）。

- A. 改邮件
- B. 编剧本
- C. 擦地板
- D. 写代码

【答案】C

【解析】本题根据题目中的资料提示：“它能够通过理解和学习人类的语言”，判断 GPT 可以完成文本相关工作和任务，C 选项擦地板不属于文本任务，其余三个选项均是文本任务。

3. Python 表达式  $2+3**2*3$  的值是（ ）。

- A. 15625
- B. 731
- C. 75
- D. 29

【答案】D

【解析】本题考察算术运算符的运算优先级，本题中优先算  $3^2=9$ ，然后算乘法， $9*3=27$ ，最后算加法， $2+27=29$ ，所以选择 D 选项。

4. Python 语句 `print(5//2*3, 5%2*3)` 执行后的输出是（ ）。

- A. 0 5
- B. 05
- C. 6 3
- D. 63

【答案】C

【解析】本题考察算术运算符和 `print` 逗号隔开连续输出，逗号前面整除和乘法是同等级，按从左向右运算，结果为 6，逗号后面部分运算结果为 3，`print` 函数输出多个默认中间使用逗号隔开，故 D 选项错误。

5. 执行 Python 语句 `a = int(input())` 时如果输入 3.14，下述说法正确的是（ ）。

- A. 变量 a 将被赋值为整数值 3
- B. 变量 a 将被赋值为浮点数 3.14
- C. 变量 a 将被赋值为整数值 4
- D. 语句执行将报错

【答案】D

【解析】本题考察 input 和 int 函数，input 接受的数字会被转换成字符串格式，所以接收的 3.14 是字符串类型，而 int 函数不能将字符串类型的小数形数字转换成浮点型数据，程序会报错，故选择 D 选项。

6. 下面 Python 代码执行后的输出是（ ）。

```
n = 18
if n % 3:
    print(0,end="#")
else:
    print(1,end="#")

if n % 5:
    print(0)
else:
    print(1)
```

- A. 0#1
- B. 1#0
- C. 0#0
- D. 1#1

【答案】B

【解析】本题考察分支语句，if 语句后方为真执行 if 下方代码块否则执行 else 下方代码块，n=18，n%3=0，0 为假，所以第一对分支语句执行 else 代码块，输出 1#。n%5=3，3 为真，所以第二对分支语句执行 if 代码块，输出 0，故最终输出 1#0，选择 B 选项。

7. 下面 Python 代码执行后的输出是（ ）。

```
tnt = 0
for i in range(-50,50):
    tnt += i
print(tnt)
```

- A. -50
- B. 0
- C. 50
- D. 100

【答案】A

【解析】本题考察循环和累加，range 函数是包含起点，不包含终点。所以范围是-50~49，范围内所有数累加时，从1~49正负互相抵消为0，加0不变还是0，只剩下一个-50，故选择A选项。

8. 下面 Python 代码执行后的输出是（ ）。

```
tnt = 0
for i in range(10,2):
    tnt += i
print(tnt)
```

- A. 0
- B. 3
- C. 20
- D. 25

【答案】A

【解析】本题考察循环和累加，range 函数第一个参数为起点，第二个参数为终点。所以本题中循环是要从大数到小数，但是没有写步长为负数，默认步长为1，向上增长，所以循环执行0次，即代码不执行该循环，相当于直接输出为0的tnt，故选择A选项。

9. 下面 Python 代码执行后的输出是（ ）。

```
tnt = 0
for i in range(2,6):
    if i % 2:
        tnt += i
print(tnt)
```

- A. 6
- B. 8
- C. 12
- D. 14

【答案】B

【解析】本题考察依然循环和累加，range 函数是包含起点，不包含终点。所以循环变量  $i$  的范围是 2~5，在 2、3、4、5 四个数中 2 和 4 取余 2 为 0，不能走 if 语句，3 和 5 取余 2 不为 0，if 为真能够执行累加操作，3 和 5 能被累加，故结果为 8，选择 B 选项。

10. 下面 Python 代码执行后的输出是（ ）。

```
n = 10
tnt = 0
while n != 0:
    if n % 3:
        tnt += n
    n -= 1
print(tnt)
```

- A. 18
- B. 22
- C. 33
- D. 37

【答案】D

【解析】本题考察循环、分支和累加逻辑，首先循环变量  $n$  的范围是从 10 到 1，其中 10、8、7、5、4、2、1 取余 3 不为 0，if 语句为真，则能执行累加语句被累加到  $tnt$  中，故结果为 37，选择 D 选项。

11. 下面 Python 代码执行后的描述，正确的是（ ）。

```
import turtle

for i in range(10):

    if i % 2 == 0:
        turtle.pencolor("red")
    else:
        turtle.pencolor("blue")

    turtle.forward(i*10)
```

- A. 有 4 条蓝色线段 5 条红色线段，总计 9 条线段
- B. 有 4 条红色线段 5 条蓝色线段，总计 9 条线段
- C. 有 5 条蓝色线段 5 条红色线段，总计 10 条线段
- D. 有 5 条红色线段 5 条蓝色线段，总计 10 条线段

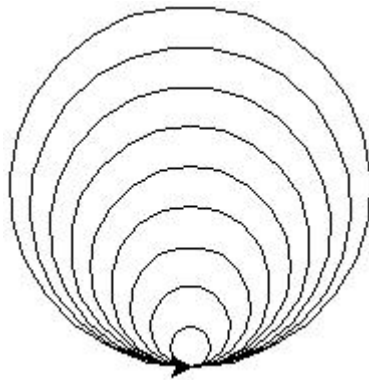
【答案】B

【解析】本题考察循环、分支和 turtle 模块，代码中循环 10 次，循环变量  $i$  范围是从 0 到 9，根据奇偶决定颜色，其中第一次循环  $i$  的值为 0， $0*i=0$ ，所以前进 0 像素，第一次循环没有线段产生，整体段数为 9。其中 1、3、5、7、9 是奇数，产生蓝色段，2、4、6、8 是偶数，产生红色段，故选择 B 选项。

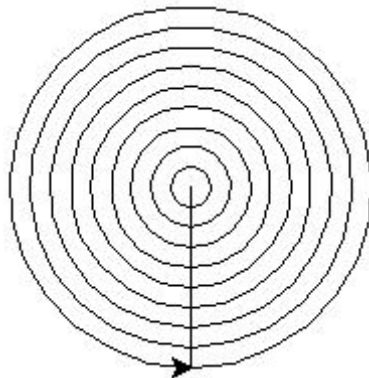
12. 下面 Python 执行后输出是（ ）。

```
import turtle

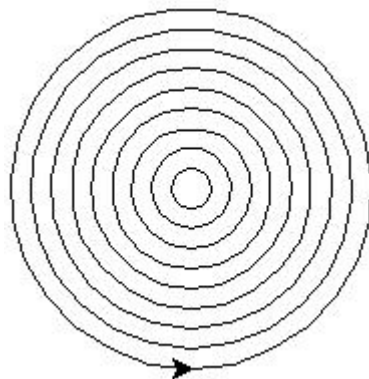
for i in range(10):
    turtle.circle(i*10)
```



A.



B.



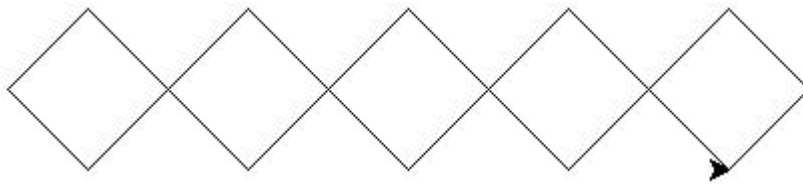
C.

D. 以上都不正确

【答案】A

【解析】本题考察循环和 turtle 模块，代码中循环 10 次，共 9 个圆，并没有画笔坐标移动，9 个圆形都是从一个位置出发，半径分别为 10、20、30..... 形成 A 选项所示图形，故选择 A 选项。

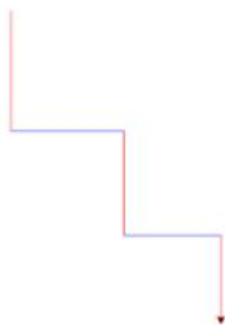
13. 下面 Python 代码执行后正确的图形是（ ）。



```
import turtle

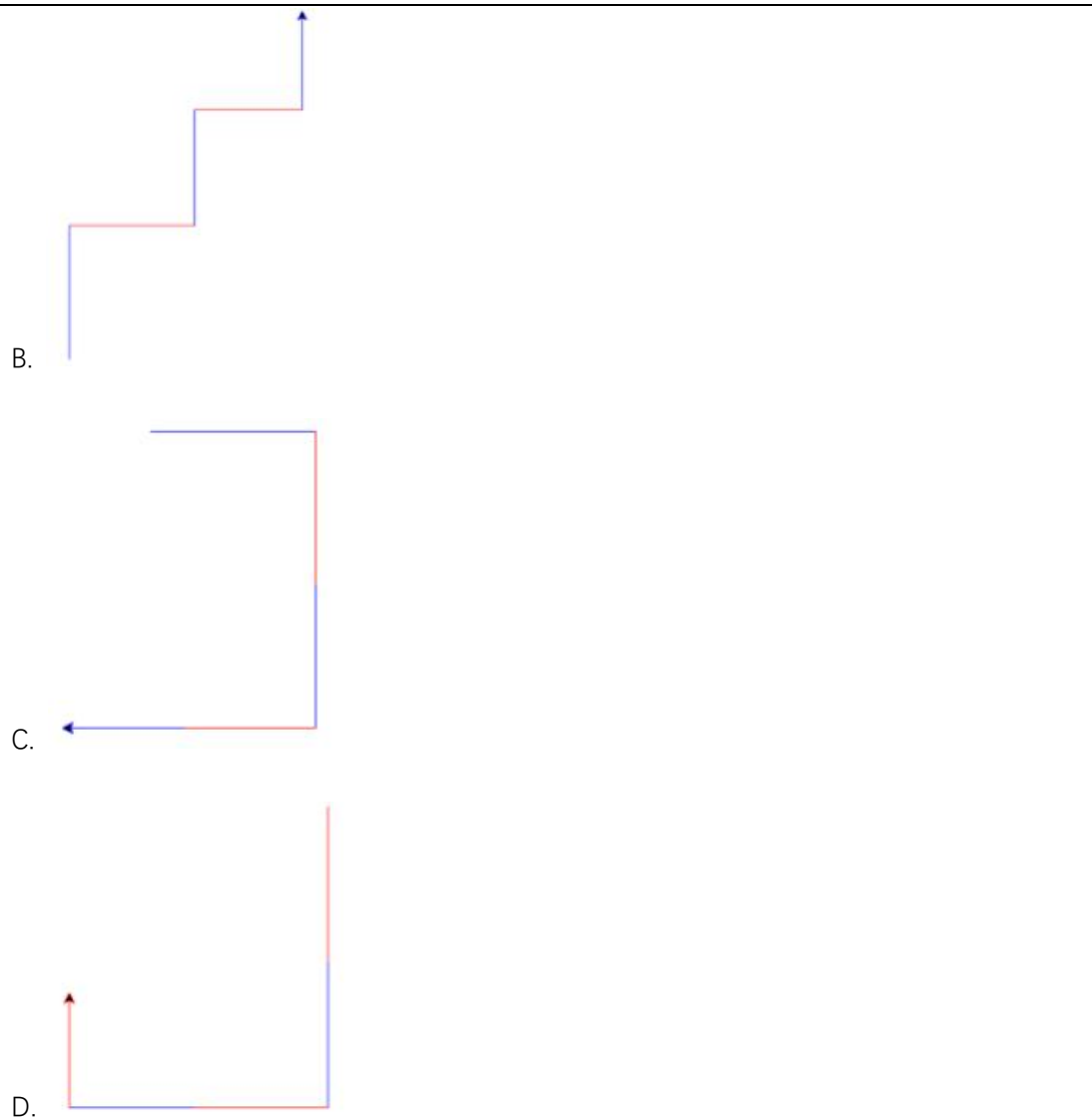
for i in range(5):
    if i % 2 == 0:
        turtle.right(90)
        turtle.pencolor("red")
    else:
        turtle.left(90)
        turtle.pencolor("blue")

    turtle.forward(10*15-10*i)
```



A.





【答案】A

【解析】本题考察循环和 turtle 模块，首先是循环次数为 5 次，根据最后的前进命令可知，每次循环前进的长度依次减小，根据循环变量 i 的奇偶性，决定转弯的方向和画笔颜色，画笔在初始朝右的状态下是先右转红色，再左转蓝色，后面以此类推，故选择 A 选项。

14. 为画出如下所示图形，下面 Python 代码横线处应填入（ ）。

```
import turtle

for i in range(5):
    turtle.penup()
    turtle.goto(_____)
    turtle.pendown()

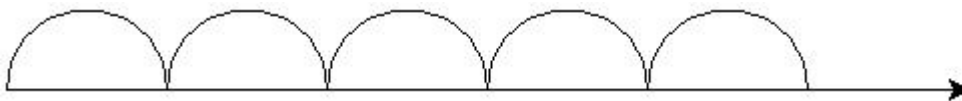
    turtle.circle(40,steps = 4)
```

- A.  $i*80,0$
- B.  $i*40,0$
- C.  $0,i*80$
- D.  $0,i*40$

【答案】A

【解析】本题考察循环和 turtle 模块，根据要求补全程序，图中的正方形是使用 circle 命令画出来的，通过循环每次 goto 到下一次的起点，由圆形的半径是 40 可以推断出每次向右移动的距离应为 80，一直处于同一高度，所以 y 坐标不变，故选择 A 选项。

15. 为画出如下所示图形，下面 Python 代码横线处应填入（ ）。



```
import turtle

for i in range(5):
    turtle.left(90)
    turtle.circle(40, _____)
    turtle.goto(80*(i+1),0)
    turtle.left(90)
```

- A. -180
- B. -90
- C. 90
- D. 180

【答案】D

【解析】本题考察循环和 turtle 模块，根据要求补全程序，图中是使用 circle 命令画半圆形，可确定圆心角是 180 度，如果角度写-180，则圆形是向下的，处于横线下方；一直处于同一高度，所以 y 坐标不变，故选择 D 选项。

## 二、判断题（每题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	√	√	√	×	×	×	√	√	×	√

1. 计算机硬件主要包括运算器、控制器、存储器、输入设备和输出设备。

【答案】正确

【解析】本题考察计算机基础知识中的组成部分，属于概念题，硬件主要包括运算器、控制器、存储器、输入设备和输出设备。

2. 诞生于 1958 年的 103 机是中国第一台通用数字电子计算机，比 1946 年在美国诞生的第一台通用电子计算机 ENIAC 晚了十多年。

【答案】正确

【解析】本题考察计算机基础知识中的计算机发展史，属于概念题，103 型通用数字电子计算机是我国第一台电子计算机，于 1958 年 8 月研制成功，平均每秒运算 1 万次，接近当时英国、日本计算机的指标。

3. 在 Python 代码中变量 n 被赋值为正整数，则表达式 `print(n % 10)` 的值为正整数 n 的个位数。

【答案】正确

【解析】本题考察运算符中的算术运算符%，正整数对 10 进行取余运算可以得到不够 10 没进位的个位数，例如： $58\%10=8$ 、 $13\%10=3$ 。

4. Python 语句 `print(2, 3, sep="#", end="->")` 中的 `sep` 和 `end` 先后顺序不能改变。

【答案】错误

【解析】本题考察 `print` 输出函数，`sep` 和 `end` 两个参数为关键字参数，调换顺序也可以通过名字，找到对应位传递位置，先后顺序改变，输出结果不变。

5. Python 函数 `input()` 可以输入字符串、整数、浮点数等。

【答案】错误

【解析】本题考察 `input` 输入函数，`input` 接收的输入会被转换成字符串，无论输入的是数字还字符串，都已字符串的格式存储，所以我们平时做题有时需要配合 `int` 进行数据类型转换。

6. Python 表达式 `int("10"*2)+10` 的值为整数 30。

【答案】错误

【解析】本题考察数据运算和数据类型转换，首先是字符串 10 乘数字 2 之后变成字符串 1010 后，被 `int` 函数转换为数字 1010，再和数字 10 做常规加法运算， $1010+10=1020$ ，故本题说法错误。

7. 在 Python 中，通常可以用 `while` 模拟实现 `for-in` 循环，但 `for-in` 未必能模拟实现 `while` 循环。

【答案】正确

【解析】本题考察 `for` 循环和 `while` 循环，这两种循环形式不同，某些情况可以相互和转换，但有些时候涉及到循环内局部变量和全局变量时未必能直接转换，故本题说法正确。

8. 在 Python 代码中，可以将变量命名为 `print`，虽然 `print` 是输出函数名称，但该变量命名非常不好。

【答案】正确

【解析】本题考察 python 中的变量命名规则，其中有规定不使用和关键字冲突的名字，虽然某些情况下可以运行通过，但这样的命名方式确实非常不好，故本题说法正确。

9. 以下 Python 代码将绘制一个红色填充的圆。

```
import turtle
turtle.fillcolor("red")
turtle.circle(20)
```

【答案】错误

【解析】本题考察 turtle 模块中的填充知识点，代码中只是设置了填充颜色，并没有使用开始填充和结束填充命令将圆形命令包裹起来，所以不会有填充颜色，故本题说法错误。

10. Python 代码 turtle.forward() 执行后，不改变海龟朝向。

【答案】正确

【解析】本题考察 turtle 模块，前进命令 forward 只是画笔位置移动，不会改变画笔当前朝向，故本题说法正确。

### 三、编程题

#### 第 1 题

【问题描述】

小明在为自己规划学习时间。现在他想知道两个时刻之间有多少分钟，你能通过编程帮他做到吗？

【输入描述】

输入 4 行，第一行为开始时刻的小时，第二行为开始时刻的分钟，第三行为结束时刻的小时，第四行为结束时刻的分钟。输入保证两个时刻是同一天，开始时刻一定在结束时刻之前。时刻使用 24 小时制，即小时在 0 到 23 之间，分钟在 0 到 59 之间。

【输出描述】

输出一行，包含一个整数，从开始时刻到结束时刻之间有多少分钟。

【样例输入 1】

9  
5

9

6

【样例输出 1】

1

【样例输入 2】

9

5

10

0

【样例输出 2】

55

参考代码：

```
h1 = int(input())
```

```
m1 = int(input())
```

```
h2 = int(input())
```

```
m2 = int(input())
```

```
print(h2*60+m2-h1*60-m1)
```

【解析】 本题考察 python 中的输入输出和数据运算，首先是输入分为 4 行，所以需要 4 个 input 函数和四个变量，又因为 input 函数接收的数据会被转换成字符串，后面需要做数字运算，所以每个输入都配合 int 函数转换成整型。最后输出的运算是将单位统一后运算，小时数乘以 60 转换为分钟，再加上正常的分钟，即：结束小时\*60+结束分钟-开始小时\*60-开始分钟，相当于用很大的分钟数减去一个很大的分钟数，算出了学习的时间。

## 第 2 题

【问题描述】

1. 累计相加，形如：

$$1+(1+2)+(1+2+3)+(1+2+3+4)+\dots+(1+2+3+4+5+\dots+n)1+(1+2)+(1+2+3)+\dots+(1+2+3+4)+\dots+(1+2+3+4+5+\dots+n);$$

2. 利用 input() 语句输入一个大于 1 的正整数；

3. 观察后一项与前一项的关系；

【输入描述】

1. 输入一个大于 1 的正整数，假设输入合规，不考虑不合规情景，如带小数点的数，负数等；
2. **特别提示：**常规程序中，输入时好习惯是有提示。考试时由于系统限定，输入时所有 input() 函数不可有提示信息。

**【输出描述】**

1. 输出累计相加的结果

**【样例输入 1】**

3

**【样例输出 1】**

10

**【样例输入 2】**

4

**【样例输出 2】**

20

**【样例输入 3】**

10

**【样例输出 3】**

220

参考程序：

#计算：1+(1+2)+(1+2+3)+(1+2+3+4)+.....+(1+2+3+4+.....+n)

```
N = int(input())
```

```
allTnt = 0 #保存全部序列之和
```

```
subTnt = 0 #保存子序列之和
```

```
for i in range(1,N+1):
```

```
    subTnt += i
```

```
    allTnt += subTnt
```

---

```
print(allTnt)
```

【解析】本题考察循环配合数值累加知识点，要求 input 函数没有提示词，所以配合 int 函数直接将输入的数据转换成整型存入 N 中，创建两个变量并赋初始值为 0，用于累加，题目要求从 1 开始，range 函数是包含开始不包含结束，所以写法为 range(1, N+1)，循环变量 i 已经符合题目要求的变化规律，subTnt 会累加 i，allTnt 累加 subTnt。也就是说每次循环 subTnt 变化规律是：1、1+2、1+2+3... 1+2+...+n。而 allTnt 是将这些又累加起来，符合题目要求，最后输出累加结果 allTnt。