

## 2023 年 GESP9 月认证 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 9 月份 Python 一级认证真题解析。

### 一、单选题 (每题 2 分, 共 30 分)

1. 我们通常说的“内存”属于计算机中的 ()。

- A. 输出设备
- B. 输入设备
- C. 存储设备
- D. 打印设备

**【答案】C**

**【解析】**本题属于考察计算机基础知识中的“输入”和“输出”概念; “输入”指外界向机器内部传递信息, “输出”指计算机内部信息向外界展示。“内存”属于是计算机的重要部件, 也称内存储器和主存储器。所以是存储设备, 选择 C。

2. 以下 Python 不可以作为变量的名称的是 ()。

- A. redStar
- B. RedStar

C. red\_star

D. red star

【答案】D

【解析】本题考察 Python 语言中变量名的定义规则：Python 中声明变量名时，变量名中能且只能包含字母、数字、下划线，且不能用数字开头，选项中 A、B、C 均满足要求，只有 D 选项 red 和 star 中间是空格，不符合要求，故选 D。

3. Python 表达式  $2 - 1 ** 0 \% 10$  的值是 ( )。

A. 0

B. 1

C. 2

D. 3

【答案】B

【解析】本题考察算术运算符的运算优先级，本题中优先算  $1^0=0$ ，然后算取余运算  $1\%10=1$ ，最后算减法  $2-1=1$ ，所以选择 B 选项。

4. 下面 Python 代码执行后的输出是 ( )。

```
a , b = 3 , 4
print("a+b=",a+b)
```

A. 3+4= 7

B. 3+4=7

C. a+b= 7

D. a+b=7

【答案】C

【解析】本题考察 Python 中的多变量赋值，算术运算符和 print 逗号隔开连续输出，首先需要了解赋值运算符的规则，如  $a,b=3,4$ ，相当于  $a=3,b=4$ ，其次是 print 函数输出，输出多个内容默认中间使用逗号隔开，输出顺序为从左向依次

输出。逗号前面是字符串输出“a+b=”，逗号后面输出是变量 a+b 的计算结果 7。所以最后输出内容为 a+b= 7。故 C 选项正确。

5. 下面 Python 代码执行后的输出是 ( )。

```
a , b = 1 , 2
a , b = b , a
print(a, b)
```

- A. 1 2
- B. 2 1
- C. 1 1
- D. 2 2

**【答案】B**

**【解析】**本题考察 Python 中的多变量赋值和 print 输出。两个 Python 变量值交换符合 Pythonic 特征，其他语言多是三角交换，所以 a, b=1, 2，相当于 a=1,b=2，其次是 a, b=b,a,相当于 a=b, b=a。所以最后 a=2, b=1，print 函数依次输出 2，1。故选 B。

6. 下面 Python 代码执行后的输出是 ( )。

```
m = 14
n = 12
if m % 2 == 0 and n % 2 == 0:
    print("都是偶数")
elif m % 2 == 1 and n % 2 ==1:
    print("都是奇数")
else:
    print("不都是偶数或奇数")
```

- A. 都是偶数
- B. 都是奇数
- C. 不都是偶数或奇数

D. 以上说法都不正确

【答案】A

【解析】本题属于考察选择语句的多分支语句和基本运算中的取余运算符、逻辑运算符 **and** 和比较运算符 **==**。首先，定义了两个变量 **m,n**，赋值为 14，12。然后使用了多分支语句进行判断，多分支语句的特点是当多个条件中有一个条件能够成立时，不再进行其他条件的判断。本题中判断条件是对应变量对 2 取余，判断是否存在余数。存在余数就是奇数，不存在余数就是偶数。这里 14，12 都是能整除 2 的偶数，对 2 取余结果为 0，**0==0** 结果为 **True**，**0==1** 结果为 **False**。**and** 左右两边结果都为 **True** 的时候结果就是 **True**，所以 if 语句中的判断条件为 **True**，故选 A。

7.下面 Python 代码执行后的输出是 ( )。

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

A.1

B.4

C.5

D.10

【答案】C

【解析】本题属于考察循环和基本运算中，题目将 for 循环和 **range()** 函数结合，实现循环加 1 的操作，**range()** 函数中只有一个参数，代表结束值，默认起始值为 0，遍历不包括结束值。**range(5)** 就是遍历 5 次，即 0，1，2，3，4。每遍历一次 **cnt** 的值加 1，**cnt** 初始值为 0，所以最后 **cnt** 为 5，故选 C。

8.下面 Python 代码执行后的输出是 ( )。



```
tnt = 0
for i in range(1, 5, 2):
    tnt = tnt + i
print(tnt)
```

- A. 2
- B. 4
- C. 6
- D. 10

**【答案】B**

**【解析】**本题属于考察循环和基本运算，题目将 for 循环和 range() 函数结合，实现循环加 i 的操作。range() 函数中有三个参数，从左到右分别代表起始值，结束值，步长。步长值就是相邻两个数之间的差值。如 range(0, 5, 2)，即从 0 开始，到 5 结束，取不到 5，步长为 2，依次取到 0, 2, 4。所以遍历循环 range(1,5,2) 得到的 i 值就是 1, 3。同时 tnt 初始值为 0，每次循环增加相应的 i，所以循环结束 tnt 为 4，故选 B。

9. 下面 Python 代码执行后的输出是 ( )。

```
n = 5
cnt = 1
while n >= 0:
    cnt += 1
    n -= 2
print(cnt)
```

- A. 3
- B. 4
- C. 6
- D. 7

**【答案】B**

**【解析】**本题属于考察 while 循环和基本算术运算。循环的条件是变量 n 大于等于 0，满足条件将 cnt 加 1，n 减少 2。题目中 n 的初始值为 5，故循环执行 3

次，再执行第四次的时候  $n$  为-1，不满足循环条件，结束循环。 $cnt$  的初始值为 1，三次循环后  $cnt$  为 4。故选 B。

10.下面 Python 代码用于输出所有能整除的数，如 10 输出成 1、2、5 和 10，12 输出为 1、2、3、4、6 和 12，17 则输出 1 和 17。在横线处应填入代码是 ( )。

```
n = input("请输入一个正整数: ")
n = int(n)

for i in _____:
    if n % i == 0:
        print(i)
```

- A. range(n)
- B. range(n+1)
- C. range(1, n)
- D. range(1, n+1)

**【答案】D**





**【解析】**本题属于考察输入语句 `input()` 函数，`for` 循环语句和 `range()` 函数的结合使用。题目要求输出  $n$  所有能整除的数，所以应遍历到 1 到  $n$  之间的每个数。`range()` 函数有两个参数时，从左到右依次是开始值，结束值，因为左闭右开原则，右侧结束数不包括，遍历过程中取不到结束值，故选择 D。

11.下面 Python 代码执行后输出的正确图形是 ( )。

```
import turtle

turtle.pencolor("RED")
turtle.pensize(5)

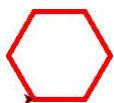
for i in range(4):
    turtle.forward(50)
    turtle.left(90)
```

- A. 
- B. 
- C. 
- D. 

【答案】B

【解析】本题属于考察 turtle 绘图库的使用和 for 循环。首先导入 turtle 库，使用了 `pencolor("RED")` 函数设置了画笔画出线条的颜色为红色。使用了 `pensize(5)` 设置画笔宽度为 5。其次，for 循环执行 4 次，每次循环执行 `forward(50), left(90)`。`forward(50)` 函数为向当前画笔方向移动 50 像素长度。`left(90)` 函数设置了左转 90 度。最后箭头应该是回到初始位置初始方向，通过看箭头的位置和方向，故选 B。

12. 为输出下面的图形，应在下面 Python 代码横线处填入代码是 ( )。



```
import turtle

turtle.pencolor("RED")
turtle.pensize(5)

for i in range(6):
    turtle.forward(50)
    turtle.left(_____)
```

- A. 30
- B. 60
- C. 90
- D. 120

【答案】B

【解析】本题属于考察 turtle 绘图库的使用和 for 循环。首先导入 turtle 库，使用 `pencolor("RED")` 函数设置了画笔画出线条的颜色为红色。使用了 `pensize(5)` 设

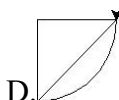
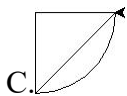
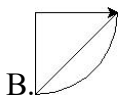
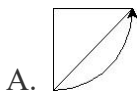
置画笔宽度为 5。其次，for 循环执行 6 次，最后实现一个六边形。因为多边形外角和等于 360 度，六边形共有六个角，因此每次要左转 60 度，故选 B。

13.下面 Python 代码执行后输出的正确图形是 ( )。

```
import turtle

turtle.left(90)
turtle.forward(80)
turtle.right(90)
turtle.forward(80)
turtle.home()

turtle.circle(80,90)
```



【答案】A

【解析】本题属于考察 turtle 绘图库的使用，forward(n)函数为向当前画笔方向移动 n 像素长度。left(n)函数设置了左转 n 度。home()设置当前画笔返回原点，即回到初始状态。circle(radius,extent)函数绘制一个 radius 指定半径的圆。圆心在海龟左边 radius 个单位；extent 为一个夹角，用来决定绘制圆的一部分。如未指定 extent 则绘制整个圆。如果 extent 不是完整圆周，则以当前画笔位置为一个端点绘制圆弧。circle(80,90)表示绘制一个半径为 80，圆心位于(0,80)，圆心角度为 90 度。故选 A。

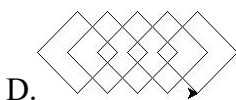
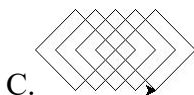
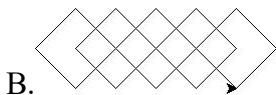
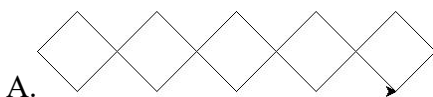


14.下面 Python 代码执行后输出的正确图形是 ( )。

```
import turtle

for i in range(5):
    turtle.penup()
    turtle.goto(i*40,0)
    turtle.pendown()

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



**【答案】B**

**【解析】**本题属于考察 turtle 绘图库的使用和 for 循环。首先导入 turtle 库，循环遍历 5 次，每次执行的代码为：使用 penup()函数将画笔抬起，使用 goto()函数让海龟移动到一个绝对坐标，不改变海龟的朝向。使用 pendown()函数实现画笔落下，在移动移动时将画线。使用 circle(radius,steps=n)函数是绘制一个 radius 半径的圆内切正 n 边形，圆是以其内切正多边形来近似表示的，其边的数量由 steps 指定。circle(40,steps=4)为对角线为 80 的正多边形。又因为每次 x 的位置增加 40，所以第一个正多边形的最右侧顶点与第二个正多边形的下顶点有相同的 x 值。应该选 B。

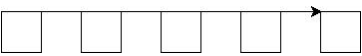
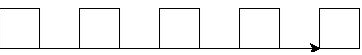
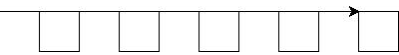

15.下面 Python 代码执行后输出的图形是 ( )。

```
import turtle

for i in range(1,10,2):

    turtle.goto(i*40,0)

    turtle.forward(40)
    turtle.right(90)
    turtle.forward(40)
    turtle.right(90)
    turtle.forward(40)
    turtle.right(90)
    turtle.forward(40)
    turtle.right(90)
```

- A. 
- B. 
- C. 
- D. 

**【答案】C**

**【解析】**本题属于考察 turtle 绘图库的使用和 for 循环。首先导入 turtle 库，循环遍历 5 次。使用了 right() 函数实现右转，所以每次循环都画一个在 X 轴下方边长为 40 的正方形。排除 D 选项，同时使用了 goto() 函数，会先绘制一段长为 40 的直线，所以 A,B 选项排除。结合代码，从 (80,0) 的位置进行第一次右转，第一个正方形绘制完成结束位置为 (40,0)，故选 C。

## 二、判断题（每题 2 分，共 20 分）

1. 早期计算机内存不够大，可以将字库固化在一个包含只读存储器的扩展卡中插入计算机主板帮助处理汉字。

**【答案】**正确√

**【解析】**本题考察计算机历史。早期的计算机使用中，因为计算机的处理能力有限，为了提高计算机的效率，减少在汉字输入过程中对存储器的频繁访问，研制

了汉卡。使用汉卡可以有效的提高计算机的速度，尽可能的减少占用计算机内存空间。但汉卡存在与其它硬件冲突的问题。随着计算机硬软件技术的进步，计算机的处理能力有了飞速的提高，因此汉卡的使用越来越少。

2. 神威·太湖之光超级计算机是中国自主研发的超级计算机，在全球超级计算机 TOP500 排行榜中多次荣膺榜首。

【答案】正确√

【解析】本题考察计算机历史。榜首意思是榜上公布的名单中的首位，泛指第一名。神威·太湖之光超级计算机是由国家并行计算机工程技术研究中心研制，安装在国家超级计算无锡中心的超级计算机。2016 年 6 月 20 日，德国法兰克福国际超算大会（ISC）公布了新一期全球超级计算机 TOP500 榜单，由国家并行计算机工程技术研究中心研制的“神威·太湖之光”以超第二名近三倍的运算速度夺得第一。之后多次荣膺榜首。

3. Python 表达式 `int("3.14")` 的值为 3。

【答案】错误×

【解析】本题考察基本数据类型中的数据类型转换的 `int()` 函数。`int()` 函数可以将字符串转换为整数，但如果字符串内是带小数的，就会出现类型转换错误。`int()` 函数要求输入数字或者整数字符，如果要将 3.14 转换为整数，可以先 `float()` 然后 `int()` 转换。故本题错误。

4. Python 语句 `print(2,3,"23")` 的输出为 2,3,23。

【答案】错误×

【解析】本题考察输出语句 `print()` 函数输出格式。`print()` 函数输出多个内容默认中间使用逗号隔开，输出顺序为从左向依次输出,中间用空格隔开，不是逗号隔开。

5. Python 内置函数 `range(10,2)` 表示从 0 开始到 10 结束但不包含 10，间隔为 2。

【答案】错误×

【解析】本题考察 `for` 循环结构中 `range()` 的使用。`range(start, end)` 函数中有两个参数时，`start` 代表开始值，`end` 代表结束值。

6. Python 表达式 `"10"*2+"10"` 的值为 `'101010'`。

【答案】正确√

【解析】本题考察 Python 字符串的基本操作。Python 中定义字符串型\*整型，代表将字符串重复 N 次，题目中的正确结果应该为 `"1010"`。Python 中定义两个字符串相加时，代表两个字符串进行拼接，所以最终为 `'101010'`。单引号或者双引号引起来的都表示字符串，故结果正确。

7. 在 Python 中，`for-in` 循环不可能导致死循环，而 `while` 有可能。

【答案】正确√

【解析】本题考察 Python 中 `for-in` 循环语句和 `while` 循环语句。死循环是一种常见的编程概念，它可以让程序永远执行下去。`for` 循环语句遍历一个可迭代的元素，当所有元素遍历完毕后，循环自动停止，`while` 循环只要循环条件为真(True)就可以一直执行循环体。

8. 在下面的 Python 代码中，由于循环中的 `continue` 是无条件被执行，因此将导致死循环。

```
for i in range(1,10):  
    continue
```

【答案】错误×

【解析】本题考察 Python 中 `for-in` 循环语句和流程控制 `continue` 关键字的掌握。死循环可以让程序永远执行下去。题目中 `i` 取值为从 1 到 9，只执行 9 次，就结束循环。故本题错误。

9. 在 Python 代码中, `turtle.home()`清除画面, 海龟回到原点, 朝向置为默认方向。

【答案】错误×

【解析】本题考察 `turtle` 绘图库中 `home()`函数的使用。`home()`函数不清除画面, 海龟回到原点, 朝向置为默认方向。`reset()`函数海龟回到原点并设置所有变量为默认值。故本题错误。

10. 在 Python 代码中, 可以用 `turtle.circle()`绘制正多边形。

【答案】正确√

【解析】本题考察 `turtle` 绘图库 `circle()`函数的使用。使用 `circle(radius, steps=n)`函数是绘制一个 `radius` 半径的圆内切正 `n` 边形, 圆是以其内切正多边形来近似表示的, 其边的数量由 `steps` 指定。故本题正确。

### 三、编程题（每题 25 分，共 50 分）

#### 1. 买文具

##### 【问题描述】

开学了, 小明来到文具店选购文具。签字笔 2 元一支, 他需要  $X$  支; 记事本 5 元一本, 他需要  $Y$  本; 直尺 3 元一把, 他需要  $Z$  把。小明手里有  $Q$  元钱。请你通过编程帮小明算算, 他手里的钱是否够买他需要的文具。

##### 【输入描述】

输入 4 行。

第一行包含一个正整数  $X$ , 是小明购买签字笔的数量。约定  $1 \leq X \leq 10$ 。

第二行包含一个正整数  $Y$ , 是小明购买记事本的数量。约定  $1 \leq Y \leq 10$ 。

第三行包含一个正整数  $Z$ , 是小明购买直尺的数量。约定  $1 \leq Z \leq 10$ 。

第四行包含一个正整数  $Q$ , 是小明手里的钱数（单位：元）。

##### 【输出描述】

输出 2 行。如果小明手里的钱够买他需要的文具，则第一行输出"Yes"，第二行输出小明会剩下的钱数（单位：元）；否则，第一行输出"No"，第二行输出小明缺少的钱数（单位：元）。

**【样例输入 1】**

1  
1  
1  
20

**【样例输出 1】**

Yes  
10

**【样例输入 2】**

1  
1  
1  
5

**【样例输出 2】**

No  
5

**【题目大意】**

判断文具总钱数与现有钱数之间的大小关系。如果文具总钱数小于于现有钱数，第一行输出"Yes"，第二行输出小明会剩下的钱数（单位：元）。如果文具总钱数大于现有钱数，输出第一行输出"No"，第二行输出小明缺少的钱数（单位：元）。

**【解题思路】**

本题主要考察输入语句，if 条件判断和比较运算符。

1. 首先输入 X,Y,Z,Q
2. 创建变量 Money 存储总钱数，即  $2 * X + 5 * Y + 3 * Z$ 。
3. 判断 Money 是否小于 Q，小于输出 Yes 和 Q-Money 的差值，否则，输出 No 和 Money-Q 的差值

**【参考程序】**

```
X=int(input())
Y=int(input())
Z=int(input())
Q=int(input())

Money=2*X+5*Y+3*Z
if Money<=Q:
    print('Yes')
    print(Q-Money)
else:
    print('No')
    print(Money-Q)
```

**2. 小明的幸运数**

**【问题描述】**

所有个位数为 k 的正整数，以及所有 k 的倍数，都被小明称为“k 幸运数”。小明想知道正整数 L 和 R 之间（包括 L 和 R）所有 k 幸运数的和，你能帮帮他吗？

**【输入描述】**

输入 3 行。第一行包含一个正整数 k，第二行包含一个正整数 L，第三行包含一个正整数 R。约定  $2 \leq k \leq 9$ ， $1 \leq L \leq R \leq 1000$ 。

**【输出描述】**

输出 1 行，符合题意的幸运数之和。

**【样例输入 1】**

7

1

10

**【样例输出 1】**

7

**【样例解释 1】**

1 和 10 之间共有 1 个 7 幸运数：7。因为 7 既是 7 的倍数，个位数又为 7。因此，结果为 7。

**【样例输入 2】**

7

10

20

**【样例输出 2】**

31

**【样例解释 2】**

10 和 20 之间共有 2 个 7 幸运数：14 和 17。14 是 7 的倍数，17 的个位数为 7。因此，结果为 31。

**【题目大意】**

找到所有个位数为  $k$  的正整数以及所有  $k$  的倍数，相加求和。

**【解题思路】**

本题主要考察输入语句，if 条件判断和取余运算符。求个位数为  $k$ ，用  $n \% 10$  即可。求倍数，用  $n \% k == 0$  即可。

1. 首先输入 luckyNum, leftNum, rightNum。
2. 创建变量 luckySum 存储总数，设定初始值为 0。
3. 遍历 leftNum 到 rightNum。
4. 判断是否是个位数为  $k$  的正整数或者  $k$  的倍数。两个条件满足其一即可
5. 满足条件 luckySum 增加相应的  $i$  值
6. 输出 luckySum





【参考程序】

```
luckyNum = int(input())
```

```
leftNum = int(input())
```

```
rightNum = int(input())
```

```
luckySum = 0
```

```
for i in range(leftNum, rightNum + 1):
```

```
    if i % luckyNum == 0 or i % 10 == luckyNum:
```

```
        luckySum += i
```

```
print(luckySum)
```