

2023 年 3 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。

1. 以下不属于计算机输入设备的有（ ）。

- A. 键盘
- B. 音箱
- C. 鼠标
- D. 传感器

【答案】B

【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。此题中，“音箱”属于向外部播放声音，故属于输出概念，正确答案为 B。

2. 计算机系统中存储的基本单位用 B 来表示，它代表的是（ ）。

- A. Byte
- B. Block
- C. Bulk
- D. Bit

【答案】A

【解析】本题属于考察计算机基础知识中，关于存储的基本单位；计算机中，表示存储大小的最小单位为“字节”，英文为“Byte”，表示一个字节存储空间大小的单位，用大写字母 B 表示，故此题选 A。

3. 下面有关 Python 的说法，不正确的是（ ）。

- A. Python 是一种面向对象的程序设计语言
- B. Python 有丰富的扩展库，是 Python 的重要特色
- C. Python 编译器可以生成目标平台的机器代码，运行效率高
- D. Python 是解释型高级语言，可以跨平台运行

【答案】C

【解析】本题考察考生对于 Python 编程环境与语言特点的了解；C 选项错误的原因：Python 是通过解释器逐行解释与运行程序，不属于编译型编程语言，且通常状况下，Python 相对于 C 或 C++ 等编程语言运行效率更低。

4. 以下选项中，符合 Python 语言变量命名规则的是()。

- A. 5Star
- B. fiveStar
- C. *Star
- D. @Star

【答案】B

【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、C、D 均不满足要求，故选 B。

5. 下面有关 Python 循环中有关 break 语句的说法，正确的是()。

- A. 含有 break 语句的所在循环，将不会执行该循环的 else 子句
- B. 含有 break 语句的所在循环，如果 break 被执行，将中断该循环；
- C. 一个循环只能有且只能有一个 break 语句；
- D. 一个循环可以有多个 break 子句，但每次执行只能一个 break 被执行；

【答案】B

【解析】本题属于考察循环结构中，流程控制关键字 break 的使用；A 选项错误在于，如果循环与 else 结合，当循环没被 break 终止时，会执行 else 中的语句；C 选项错误在于，循环结构中可以设置多个 break 语句；D 选项的每次执行可能一个 break 也不被执行，而是正常结束。

6. 在 Python 中实现 123 除以 10 分别得到其商 a 和余数 b 的代码是()。

- A. a , b = 123 % 10 , 123 // 10
- B. a , b = 123 // 10 , 123 % 10
- C. a , b = 123 / 10 , 123 % 10
- D. a , b = 123 / 10 , 123 // 10

【答案】B

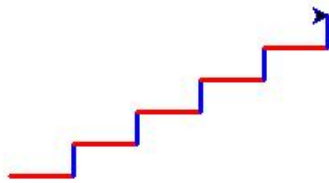
【解析】本题属于考察 Python 中的基本运算；首先需要了解赋值运算符的规则，如 a, b = 1, 2，相当于 a=1, b=2；其次考察求商运算符//和求余运算符%的使用；综上，本题正确答案为 B。

7. 下面有关 turtle.color() 的说法，正确的是（ ）。
- A. turtle.color() 函数不能没有参数；
 - B. turtle.color(1,2,3) 表示画笔颜色为 1、背景颜色为 2, 前景颜色为 3；
 - C. turtle.color("red","green") 表示画笔颜色为 red、填充颜色为 green；
 - D. turtle.color((1,2,3),(4,5,6)) 分别表示画笔颜色和填充颜色；

【答案】C

【解析】本题属于考察 Python 中 turtle 绘图库的使用；turtle 库中的 color 函数，默认颜色为黑色，故 A 错误；当接收整数作为参数时，三个整数分别表示红、绿、蓝三种颜色，参数可取值只有 0 和 1，故 B、D 错误；当接收字符串作为参数时，两个参数分别表示画笔颜色和填充颜色，故此题选 C。

8. 下面 Python 代码执行后输出如下图形，请在第 4 行 range 后括号内横线上填入正确的代码（ ）。



```
import turtle

turtle.pensize(3)
for i in range(_____):
    if i%2==0:
        turtle.pencolor("red")
        turtle.forward(40)
        turtle.left(90)
    else:
        turtle.pencolor("blue")
        turtle.forward(20)
        turtle.right(90)
```

- A. 9
- B. 10
- C. 1,9
- D. 1,10

【答案】B

【解析】本题主要考察 for 循环中变量 i 的取值范围；观察给出的图形可发现，图形中画出了 5 条红色线条和蓝色线条，程序共需要运行 10 次；通过对于程序代码的分析，当 i%2==0 表示 i 的取值为偶数时条件成立；所以第一条画出红色线条，i 的初始值应为偶数，故排除 A、C、D，正确答案为 B。

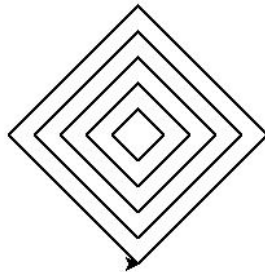
9. turtle.speed(n)用于确定绘图速度，当 n 值为（ ），速度最快。

- A. 0
- B. 1
- C. 5
- D. 10

【答案】 A

【解析】 本题考察了 turtle 绘图库中 speed() 函数；参数的取值范围为 0~10，其中参数从 1~10 画笔速度不断增加，参数为 0 时速度最快，本题属于固定用法，故选 A。

10. 下面 Python 代码执行后显示图形如下，turtle.circle() 实现，内外层半径相差为 20 像素，在横线上填入相应代码（ ）。



```
import turtle

turtle.pensize(2)
for i in range(5):
    turtle.penup()
    turtle.goto(0,-i*20)
    turtle.pendown()
    turtle.circle(_____,steps=4) #在此处填入代码

turtle.done()
```

- A. (i+1)*20
- B. i*20
- C. 20
- D. 40

【答案】 A

【解析】 本题考察了 turtle 绘图库中 circle() 函数的使用与 for 循环中变量 i 的取值；通过阅读程序发现，goto() 函数中的 y 坐标在不断变小，画笔向下运动，可以分析出正方形要逐渐放大；for 循环中，变量 i 的取值从 0 开始不断增加 1，所以可以结合变量让 circle 函数的半径变大，实现画出更大的正方形，排除 C、D 选项；同时需注意 i 初始值为 0，排除 B 选项；故正确答案为 A。

11. 下面 Python 执行后的输出结果是()。

```
n = 30

if n > 10:
    print("A")
elif n > 20:
    print("B")
elif n == 30:
    print("C")
else:
    print("D")
```

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

【答案】 A

【解析】 本题主要考察 Python 中选择结构的使用；if...elif...else... 结构，代表“如果... 否则如果... 否则...”，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进行判断；当变量 n 的值为 30 时，满足第一个 n>30 的条件，直接输出对应的结果“A”后结束这个选择结构，故正确答案为 A 选项。

12. 以下 Python 代码执行后输出结果是()。

```
Sum=0
for i in range(1,10,2):
    if i%3:
        continue
    Sum += i
print(Sum)
```

- A. 12
- B. 13
- C. 18
- D. 23

【答案】 A

【解析】 本题考察了 for 循环中 range() 函数参数与变量 i 的关系，与循环结构中流程控制关键字 continue 的使用；range() 函数的三个参数分别表示变量 i 能够取值的初始值、结束值与步长；如 range(a, N, x) 表示变量从 a 开始，到 N-1 结束，每次增加 x；由此得出题目中 i 的取值分别为 1、3、5、7、9。关键字 continue 的作用为，终止当次循环并重新开始下一次循环；条件 i%3 表示当 i 除 3 的余数非 0 时，略过当次循环，所以 Sum 中累加的 i 的值为 3 和 9，故答案为 A 选项。

13. 下面 Python 代码用于求 $\frac{1}{1} + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + \frac{1}{15} + \dots$ 之和，其中第 N 项的分母为 $1+2+\dots+N$ 之和，在横线填上合适代码以实现()。

```
#求和: 1/1+1/3+1/6+1/10+...

lastSum = 0
N = 1000000
runEnd = 0
for i in range(1,N+1):
    _____
print(runEnd)
```

- A. lastSum = i
- B. lastSum = sum(i)
- C. lastSum += i
- D. lastSum *= i

【答案】C

【解析】本题主要考察 for 循环中变量的取值，与复合赋值运算符的使用；首先需清楚 $a+=1$ 的程序写法，等效于 $a=a+1$ ；根据题意，分母应为 1 到 N 的和，阅读代码发现，i 此时的取值范围恰好为从 1 到 N，所以应该将变量 i 的值累加到 lastSum 变量中，故选项 C 为正确答案。

14. 下面 Python 代码执行后输出是 () 。

```
i=2
for i in range(i+2):
    print(i,end="#")
    i+=2
print(i)
```

- A. 0#1#2#3#4
- B. 0#1#2#3#5
- C. 0#4#4
- D. 因为 i 一直在变大，所以无穷循环永远输出

【答案】B

【解析】本题考察考生对于 for 循环中 range() 参数的设置与变量 i 取值的理解；首先 i 初始值为 2，range(i+2) 表示程序会重复执行 4 次；其次 i 的取值从 0 开始每次递增 1，所以 i 值依次取 0、1、2、3，程序输出 0#1#2#3#；最后当 i 取值为 3 的时候，变量经过 $i+=2$ 变为 5，当循环结束输出 i 时，输出数字 5，故选 B。

15. 执行以下 Python 代码后，数据结果是()。

```
Sum=0
for i in range(10):
    if i%2==0:
        continue
    if i%7==0:
        break
    Sum+=i
print(Sum)
```

- A. 0
- B. 9
- C. 10
- D. 18

【答案】 B

【解析】 本题考察考生对于 for 循环中变量 i 取值的理解和流程控制 break、continue 关键字的掌握；首先 i 取值为从 0 到 9；其次条件 1 表示当 i 为 2 的倍数时，略过当次循环，条件 2 表示当 i 为 7 的倍数时，终止整个循环；所以 Sum 变量中累加的数值依次为 0、1、3、5，最终输出的结果为 9，故此题选 B。

二、判断题 (每题 2 分, 共 20 分)

1. 在 Windows 系统中通过键盘完成对选定文本移动的按键组合是先 Ctrl+X，移动到目标位置后按 Ctrl+V。

【答案】 正确 ✓

【解析】 本题考察计算机基础中快捷键的使用；Ctrl+X 表示剪切文件(复制并从原位置删除)，Ctrl+V 表示粘贴所剪切的文件。

2. 程序员用 C、C++、Python、Scratch 等编写的程序能在 CPU 上直接执行。

【答案】 错误 ✗

【解析】 本题属于对计算机历史中，编程语言特点的考察；题目中所列举的编程语言，均无法在 CPU 上直接执行，而是需要通过各自的方式，转换成机器语言后才能被 CPU 识别和执行，故本题错误。

3. Python 代码 turtle.circle() 可以绘制每边边长相等的多边形。

【答案】 正确 ✓

【解析】 本题考察了 turtle 绘图库中 circle 函数的使用；circle 函数可以通过设置 steps 参数，来画出圆形的内接正多边形，如 circle(100, steps=3) 可以画出半径 100 的圆形中的内接正三角形，故此题正确。

4. Python 代码 turtle.goto() 执行后不会改变海龟的朝向。

【答案】 正确 ✓

【解析】 本题考察了 turtle 绘图库中 goto 函数的使用；goto 函数只会改变画笔所在的位置，并不会调整画笔的方向，故此题正确。

5. Python 的 int() 函数可以将数字式字符串形如 "3.14" 或浮点数如 3.14 转换为整数。

【答案】 错误 ✗

【解析】 本题主要考察 Python 中的数据类型概念和类型转换函数的使用；int 函数可以将只包含整数的字符串类型数据，转换为整型数据，无法将包含小数点的数据转换为整数，故此题错误。

6. Python 表达式 "10" * 2 的值为 20。

【答案】 错误 ✗

【解析】 本题主要考察 Python 中不同数据类型进行运算时的规则；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"。

7. Python 表达式 5 * 3 // 2 的值为 8，因为 15 除以 2 的商为 7.5，四舍五入即为 8。

【答案】 错误 ✗

【解析】 本题主要考察 Python 基本运算中算术运算符的使用；// 表示求除法运算中的商，所以 5*3//2 应该为 15//2，即 15 除 2 的商，结果为 7，故此题错误。

8. 在 Python 语言中，判断语句 if 可以有多个 elif 从句，但最多只能有一个 else 从句。

【答案】 正确 ✓

【解析】 本题主要考察 Python 中选择结构中的 if...elif...else... 结构；本结构可以通过 elif 设置多个“否则如果”的条件，当所有条件不满足时，程序会执行 else 中的语句，此结构中只能有一个 else，故此题正确。

9. 在 Python 中，假如 N 为正整数，则 range(N) 与 range(0, N) 等效。

【答案】 正确 ✓

【解析】 本题考察了 for 循环结构中，range() 的使用；range() 函数默认从 0 开始，故 range(N) 与 range(0, N) 效果相同。

10. Python 代码 `print("19" + "49")` 执行后将输出 1949。

【答案】 正确 ✓

【解析】 本题考察了基本数据类型中字符串的拼接操作；Python 中定义两个字符串相加时，代表两个字符串进行拼接，故此题正确。

三、编程题 (每题 25 分, 共 50 分)

1. 分数交错加减计算

【问题描述】

分数交错加减计算，形如： $\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \dots$

1. 利用 `input()` 语句输入一个大于 1 的正整数；
2. 计算规则：分母为连续奇数，分子为 1，偶数位置相减，奇数位置相加，一直计算到最接近 N（含）的奇数分母为止；
3. 计算结果四舍五入保留 8 位小数，仅输出计算结果，没有其他。

【输入描述】

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。

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】

输出四舍五入保留 8 位小数。

特别提示： 仅输出 8 位小数，没有其他。

【样例输入 1】

100

【样例输出 1】

0.78039866

【样例输入 2】

1000

【样例输出 2】

0.78489816

【题目大意】多项式求和问题，多项式中的每一项分子都为1，每一项分母都比前一项大2，累加(或减掉)每项求出整个多项式的和。

【解题思路】

1. 分析出分子、分母的规律：分子都为1，分母从1开始逐次增加2；
2. 借助 for 循环，列举 N 个多项式的值，利用 for 循环中 i 的规律表示分母；
3. 设置变量，记录多项式的项数，并设置条件判断项数为奇数或偶数；
4. 创建变量，将 N 个多项式的值依次在变量中增加或减小；
5. 按要求完成输入和输出。

【样例程序】

```
Sum = 0          #设置变量 Sum 存放多项式的和，初始值为 0
stepCount = 1   #记录多项式项数，初始值为 1
N = int(input())#输入项数总数 N
for i in range(1, N+1, 2): #控制 i 值(分母值)为递增的奇数
    if stepCount%2 == 0:  #项数为偶数时，从 Sum 中减去该项的值
        Sum -= 1/i
    else:                  #项数为奇数时，在 Sum 中增加该项的值
        Sum += 1/i
    stepCount += 1        #项数增加 1
print( round( Sum , 8) ) #使用 round 函数设置指定位数，并输出结果
```

2. 鸡兔同笼

【问题描述】

利用 `input()` 语句先后输入两个正整数，分别代表鸡兔的头和脚的总数。其中，兔有四只脚，鸡有两只脚，都只有一个头。

请编写 Python 代码计算出兔和鸡的数量，输出结果形如 “Chicken=15 Rabbit=5”，如没有解则输出 “No solution!”。

【输入描述】

分两次输入，第一次输入头的总数，回车后输入脚的总数。

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】

如果有解，则输出鸡和兔分别数量，形如 “Chicken=15 Rabbit=5” Rabbit 之前有 1 个英文半角空格，C 和 R 都为大写。

如没有解则输出 “No solution!”，No 之后有 1 个英文半角空格，solution 之后有英文半角叹号。

特别提示：注意字母大小写以及空格、叹号等匹配一致。

【样例输入 1】

20

50

【样例输出 1】

Chicken=15 Rabbit=5

【样例输入 2】

20

30

【样例输出 2】

No solution!

【题目大意】鸡兔同笼问题，输入鸡兔同笼问题中头和脚的总数，计算并输出鸡和兔子各自的数量，如果存在无解情况，输出无解提示。

【解题思路】

1. 利用循环列举所有鸡和兔子只数的可能性；
2. 根据鸡和兔子头、脚数量的特点，设置条件并判断是否满足条件；
3. 设置变量记录是否找到满足条件的解；
4. 按要求完成输入和输出。

【样例程序】

```
totalHead = int(input()) #输入头的总数
totalFoot = int(input()) #输入脚的总数
answerCount = 0 #设置变量表示是否有解，初始值为0
for chickenHead in range(totalHead+1):    #列举所有鸡的数量的可能性
    #根据鸡的数量求出兔子的数量
    rabbitHead = totalHead - chickenHead
    #判断脚的数量是否正确
    if rabbitHead * 4 + chickenHead * 2 == totalFoot:
        #按要求输出正确的解
        print(f"Chicken={chickenHead} Rabbit={rabbitHead}")
        answerCount += 1    #找到正确解后更改变量
if answerCount == 0:    #当变量值没变化时，证明没有找到正确的解
    print("No solution!")    #按要求输出无解提示
```