

2023年6月 GESP Python 二级试卷解析

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 6 月份 Python 二级认证试卷真题解析。

一、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	D	D	D	C	A	D	A	B	C	D	C	A	A	A	C

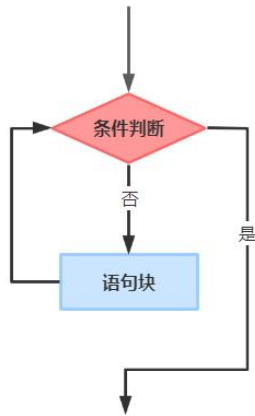
1、高级语言编写的程序需要经过以下（ ）操作，可以生成在计算机上运行的可执行代码。

- A. 编辑
- B. 保存
- C. 调试
- D. 编译

【答案】D

【解析】本题考察考生对程序设计语言中高级语言特点的了解，编译是把高级语言转换成可执行代码，使得程序能够在计算机上运行的过程；调试是指在程序开发过程中，找出并修复错误和问题的过程。

2、能够实现下面流程图功能的伪代码是（ ）。



- A. if 条件判断 then 语句块
- B. if 条件判断 then 什么也不做 else 语句块
- C. while 条件判断 do 语句块
- D. while not 条件判断 do 语句块

【答案】D

【解析】本题主要考察流程图的使用，图中表示的是条件判断，成立的时候不执行，而不成立时执行语句块，并且执行完语句块后再次进行条件判断，这是一个循环的过程，而 A 选项，表示的是如果...那么...结构，B 选项是如果...那么...否则结构，C 选项是循环，但是是条件成立的时候执行语句块，和流程图正好相反，所以本题选择 D 选项。

3、在 Python 代码中，变量 a 的值为 4，则表达式 $5 < a < 20$ 的值是（ ）。

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

【答案】D

【解析】本题考察的是一种组合比较运算符的使用，用于判断变量 a 的值是否在 5-20 之间，如果在结果为 True，不在结果为 False，需要注意的是，布尔值 True 和 False 的首字母大写。这里变量 a 的值为 4，是不在 5-20 之间的，所以结果为 False，D 选项。

4、Python 表达式 $\text{round}(8/3, 3)$ 的值是（ ）。

- A. 2
- B. 3
- C. 2.667
- D. 2.666

【答案】C

【解析】本题考察的是数学函数之四舍五入 `round()` 函数的使用，`round` 函数里有两个参数，第一个参数是需要四舍五入的数字，第二个参数是保留的小数位数（默认为 0，表示保留到整数位）。本题中对 $8/3$ 的结果保留三位小数， $8/3=2.6666666666666666\dots$ ，保留三位的结果是 2.667，故此题选 C。

5、下面 Python 代码执行后输出是（ ）。

```
import math
print(math.floor(-5.67))
```

- A. -6
- B. -5
- C. 0
- D. 报错

【答案】A

【解析】本题考察的是 python 数学库中的向下取整 `floor()` 函数，它的作用是返回不大于指定数字的最大整数。这个题目中是对 -5.67 向下取整，不大于 -5.67 的最大数字为 -6，故此题选 A。

6、下面 Python 代码执行后的输出是（ ）。

```
a, b = 3, 2
if a > b:
    a = b
    b = a
print(a,b,sep=",")
```

- A. 3,2
- B. 2,3
- C. 3,3
- D. 2,2

【答案】D

【解析】本题首先考察的是 python 下多重赋值的一种方法，用于同时给多个变量赋值，本题中，左边的变量 a 和 b 分别和右边的元素进行匹配赋值，即 a 被赋值为 3，b 被赋值为 2。然后考察分支语句中的 if 语句，表示如果...，当 if 后的条件为 True 时，if 语句下面缩进的代码块将被执行，本题中 a>b 是成立的，所以会执行下面的两行代码，变量 a 被重新赋值为 b，即 a=2，然后 b=a，此时 a 的值为 2，所以 b=2。最后通过 print() 函数输出出来，添加了 sep 参数，所以 a 和 b 之间的值会以逗号隔开，故此题选 D。

7、下面 Python 代码执行后的输出是（ ）。

```
m, n = 13, 3
z = 13 % 5
if z < 5:
    print(5)
elif z < 4:
    print(4)
elif z < 3:
    print(3)
else:
    print("0-2")
```

- A. 5

- B. 4
- C. 3
- D. 0-2

【答案】 A

【解析】 本题主要考察 Python 中多分支结构的使用，`if...elif...else...` 结构，表示“如果... 否则如果... 否则...”，这种结构可以根据不同条件的真假选择执行不同的代码块，不过，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。本题中变量 $z=13\%5$ ，%表示取余，那么 $z=3$ ，满足第一个条件 $z<5$ ，执行 `print(5)` 后结束判断，故正确答案为 A 选项。

8、下面 Python 代码执行后的输出是（ ）。

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

- A. 15
- B. 10
- C. 5
- D. 0

【答案】 B

【解析】 本题考察了考生对 python 中 for 循环嵌套结构的理解，在每一次外层循环执行过程中，内层循环会完整地执行一次。本题中，变量 `cnt` 用来计数，通过双层循环来累加其值，外层循环 `i` 的范围是从 0 到 4，每次内层循环 `j` 的范围是从 0 到 `i-1`。执行过程如下：

- (1) 当 `i` 等于 0 时，内层循环不会执行；
- (2) 当 `i` 等于 1 时，内层循环执行一次，`j` 的范围是 `range(1)`，即 0，所以 `cnt` 的值增加 1；
- (3) 当 `i` 等于 2 时，内层循环执行两次，`j` 的范围是 `range(2)`，即 0 和 1，所以 `cnt` 的值增加 2；

(4) 当 i 等于 3 时，内层循环执行三次， j 的范围是 `range(3)`，即 0、1 和 2，所以 `cnt` 的值增加 3。

(5) 当 i 等于 4 时，内层循环执行四次， j 的范围是 `range(4)`，即 0、1、2 和 3。所以 `cnt` 的值增加 4；

因此，最后 `cnt` 的值为 $1+2+3+4=10$ 。

- 9、求 m 到 n 之间的间隔数数量， m 可能大于 n ，如 3 到 6 之间间隔数数量为 2，6 和 3 之间的间隔数同样是 2。根据上述题意，在 Python 代码横线处应填上（ ）。

```
#求m-n之间的间隔数量，如3到6之间间隔数量是2
m = int(input("请输入起点数:"))
n = int(input("请输入终点数:"))

gap = 1 #设定间隔值
if m > n :
    _____

cnt = 0
for i in range(m+1, n, gap):
    cnt += 1
print(cnt)
```

- A. `gap = 1`
- B. `gap = 0`
- C. `gap = -1`
- D. `gap = gap`

【答案】 C

【解析】 本题考察了考生对 `for` 循环中 `range()` 函数中的三个参数的理解，这三个参数分别表示起始值、终止值和步长，取出的数字从起始值开始，到终止值-1 结束，每次迭代增加或减少步长的值。当步长为正数时，起始值必须小于终止值；而当步长为负数时，起始值必须大于终止值，否则，循环无法进行。在本题中，`for` 循环中的起始值为 $m+1$ ，终止值为 n ，步长为 `gap`，当 $m > n$ 时，步长 `gap` 是负数才可以取到值，所以本题选择 C 选项。

- 10、正整数 N 除以 16 的余数为 0-15，将 0-15 对应为不同字符，0-9 仍然为 0-9,10-15 依次对应为大写应为字母 A-F。根据上述题意，在 Python 代码横线处应填上（ ）。

```
N = int(input())
Remainder = N % 16 #保存余数
if N % 16 < 10:
    print(str(Remainder))
else:
    print(_____)
```

- A. chr(ord("A")+Remainder-10)
- B. chr(ord("A")+Remainder)
- C. chr(Remainder)
- D. chr(ord("A")+Remainder-10)

【答案】D

【解析】本题主要考察的是 ASCII 码和字符之间的转换，ord() 函数用于返回给定字符的 ASCII 值或 Unicode 值，括号里的参数是一个字符，返回值为该字符对应的整数值。例如，ord('A') 返回 65，ord('a') 返回 97。而 chr() 函数则相反，括号里的参数是一个整数值，返回值为该整数值对应的字符。例如，chr(65) 返回字符 'A'，chr(97) 返回字符 'a'。

本题中，N 是用户输入的正整数，Remainder 是 N 除以 16 的余数，根据题目要求，如果余数小于 10，即为 0-9 时，直接打印该余数（转换为字符串形式），否则需要将余数转换为对应的大写字母 A-F。

D 选项 chr(ord("A")+Remainder-10) 中，先将大写字母 A 的 ASCII 值通过 ord("A") 获取到，然后加上余数，比如当余数是 11 时，对应的字母是 B，和 A 相比，ASCII 值差了一个 1，所以加上余数 11 之后，又减掉了一个偏移量 10，这样再通过 chr() 函数就可以得到对应的字符。

- 11、下面 Python 代码执行后的输出是（ ）。

```
cnt = 0
for i in range(10):
    for i in range(i):
        if i % 2:
            break
        cnt += 1
print(cnt)
```

- A. 55
- B. 18
- C. 9
- D. 33

【答案】C

【解析】本题主要考察的是考生对双层 for 循环的嵌套以及 break 语句的理解。这道题目中有两个嵌套的 for 循环，外层循环的变量为 i，范围为 0 到 9，i 每取到一个值时，进入到里层循环，而里层循环的变量也为 i，范围为 range(i)，即 0 到 i-1，在每次内层循环中，我们首先检查条件 if i % 2，即判断 i 是否为奇数，如果是奇数，那么我们执行 break 语句跳出内层循环，不是奇数时，我们执行 cnt += 1 将 cnt 的值增加 1。

- (1) 外层循环中，第一次 i=0，内层循环不会执行；
- (2) 外层循环中，第二次 i=1，进入到里层循环，里层 i 的范围是 range(1)，即 0，此时 i 不是奇数，执行 cnt+=1。执行完 cnt=1；
- (3) 外层循环中，第三次 i=2，进入到里层循环，里层 i 的范围是 range(2)，即 0 和 1。i 等于 0 时不是奇数，执行 cnt+=1。下一轮里层 i 等于 1，是奇数，执行 break 语句跳出内层循环。执行完 cnt=2；
- (4) 外层循环中，第四次 i=3，进入到里层循环，里层 i 的范围是 range(3)，即 0、1 和 2。i 等于 0 时不是奇数，执行 cnt+=1。下一轮里层 i 等于 1，是奇数，执行 break 语句跳出内层循环。执行完 cnt=3；
- (5) 外层循环中，第五次 i=4，进入到里层循环，里层 i 的范围是 range(4)，即 0、1、2 和 3。i 等于 0 时不是奇数，执行 cnt+=1。下一轮里层 i 等于 1，是奇数，执行 break 语句跳出内层循环。执行完 cnt=4；

.....

后面每次都是如此，所以最终结果 cnt=9，所以本题选择 C 选项。

12、 有关下面 Python 代码执行的说法，正确的是（ ）。

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

- A. 代码执行后如果输入 5，则将输出 5
- B. 代码执行后如果输入 5，则将再输入 5 次，输出由先后输入值决定
- C. 代码输入执行后如果不输入小于 0 的数，将无限循环
- D. 代码执行将报错。

【答案】 A

【解析】 本题考查考生对 for 循环中 range() 参数的理解。程序运行后，会等待用户输入一个值，输入完成后，会将输入的数字转换为整数类型，并把这个整数作为参数传递给 range() 函数，用于创建一个范围对象，此时 for 循环的范围就确定下来了，之后变量 i 会依次取范围对象中的值，从 0 开始递增，直到范围对象的最大值（用户输入的值-1）为止。每次循环中，cnt += 1 语句会执行一次，即将 cnt 的值增加 1。这样实现了一个根据用户输入的值，执行循环并统计循环的次数的效果。

A 和 B 选项中，输入 5，范围确定即为 range(5)，执行 5 次，最后输出 5，所以 A 选项正确，B 选项错误。C 选项中，不输入小于 0 的数字，即输入大于等于 0 的数字时，输入的是几，执行几次，输出的也是几，所以 C 选项也错误。

13、 两个正整数的最大公约数是指能被两个数都整除且最大，如 12 和 18 能被 2、3 和 6 整除，但 6 最大，所以 12 和 18 的最大公约数是 6。两个质数如 13 和 17 的最大公约数是 1。根据上述题意，在 Python 代码横线处应填上（ ）。

```
m = int(input())
n = int(input())
if m > n :
    m, n = n, m
for i in _____:
    if m % i == 0 and n % i == 0:
        print(f"最大公约数={i}")
        break
```

- A. range(m, 0, -1)
- B. range(m)
- C. range(m, 1, -1)
- D. range(1, m)

【答案】 A

【解析】 本题考查的是考生对最大公约数的理解和 for 循环 range() 参数的设置以及 break 语句的作用。这段代码求最大公约数，指的是能被两个数都整除且最大的那个数，根据代码，是找到一个能够整除的数之后执行 break 语句，即终止了循环，此时找到的一定是一个最大的公约数，说明是倒着取数，所以范围是从 m 到 0 (0 取不到)，并且步长为-1。C 选项中范围是从 m 到 1，此时 1 取不到，所以本题选择 A 选项。

14、 产生 m 和 n 之间的随机整数，包括 n 和 m。根据上述题意，在 Python 代码横线处应填上 ()。

```
import random
m = int(input())
n = int(input())
if m > n:
    m, n = n, m
print(_____)
```

- A. `m + 1 + int((n - m) * random.random())`
- B. `m + 1 + int((n - m) * random.random)`
- C. `m + 1 + int((n - m)) * random.random`
- D. `m + 1 + (n - m) * int(random.random())`

【答案】A

【解析】本题考查考生对 random 随机库以及题目的理解，`random.random()` 生成一个 $[0, 1)$ 范围内的随机小数，B 选项和 C 选项没有括号，所以错误；D 选项中 `int(random.random())` 的结果始终为 0，乘以 $(n-m)$ 结果还是 0，再加上 $m+1$ ，取到的值始终是输入的两个数中较小的数加 1，所以 D 选项错误；A 选项中，先计算 n 和 m 的差值 $(n - m)$ ，然后使用 `random.random()` 生成一个 $[0, 1)$ 范围内的随机小数。将这个随机小数乘以差值得到一个 $[0, n - m)$ 范围内的随机小数，再将其转换为整数类型，最后，将得到的随机整数与 $m + 1$ 相加，便可以得到一个在 m 和 n 之间的整数，所以本题选择 A 选项。

15、下面 Python 代码执行后输出“OK”（ ）行。

```
for i in range(8, 2, -2):
    for j in range(i):
        print("ok")
```

- A. 0
- B. 12
- C. 18
- D. 20

【答案】C

【解析】本题主要考察考生对 for 循环中 `range()` 参数以及 for 循环中变量的范围的理解，外层循环中范围是 `range(8, 2, -2)`，表示从 8 开始，到 2 的前一个数结束，每取一个数字减 2，即依次取出 8、6 和 4；里层循环中 `range()` 的参数有一个，为 i ，即依次为 8、6 和 4，表示分别执行了 8 次、6 次和 4 次，共 18 次，所以输出 OK 18 行，所以本题选择 C 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	X	X	X	X	√	X	X	X	√	X

1. 诞生于 1986 年的中华学习机 CEC-I 入选了 2021 年的 CCF 计算机历史记忆（一类），它的内存只有 64KB。当时的汉字编码字符集 GB2312 中共有 6763 个汉字，假如每个汉字用 2 个字节编码，将整个 GB2312 汉字字符集都放入 CEC-I 的内存，也只占用了不超过 1/5 的内存空间。

【答案】 错误

【解析】 本题主要考察计算机基础知识的内容。1KB=1024 字节，那么 64KB = 64 * 1024 = 65536 字节，而 6763 个汉字，每个汉字占用 2 个字节，总共需要的字节数为 6763 * 2 = 13526 字节，占用的内存空间超过了 1/5 的内存空间。

2. 域名是由一串用点分隔的名字来标识互联网上一个计算机或计算机组的名称，CCF 编程能力等级认证官方网站的域名是 gesp.ccf.org.cn，其中顶级域名是 gesp。

【答案】 错误

【解析】 本题主要考查计算机基础的知识。顶级域名指的是最右边的字符组，通常用于表示国家、地区、组织或类型等信息。在给定的域名 “gesp.ccf.org.cn” 中，顶级域名是 “cn”，表示中国。所以本题错误。

3. Python 表达式 5.0//2 的值为整数 2。

【答案】 错误

【解析】 本题考查 Python 算术运算符中 // 运算符的使用。// 表示向下取整，当 // 左右都是整数时，结果是一个整数类型，而当 // 左右有一个浮点数时，结果为浮点数类型。5.0//2 的值为浮点数 2.0，所以本题错误。

4. 执行 Python 代码 a=f' {101:2.2f}' 后，变量 a 的类型为 float，其值 101.00。

【答案】 错误

【解析】 本题考查格式化字符串的内容。f' {101:2.2f}' 使用的是 f-string 格式化字符串的方法，结果是一个字符串类型，不是 float。所以本题错误。

5. 当 Python 执行循环时，如果执行到 break 语句，这终止其所在循环。

【答案】 正确

【解析】 本题考查考生对 break 语句的理解。在循环执行过程中，遇到 break 语句时，会立即终止所在的循环，不再执行循环内的剩余代码，而是跳出循环，继续执行循环后面的代码。所以本题正确。

6. 执行以下 Python 代码，如果 n 值为负整数，将输出 0。

```
n = int(input())#假设输入为负整数、零和正整数
cnt = 0
while n:
    n -= 1
    cnt += 1
print(cnt)
```

【答案】 错误

【解析】 本题主要是对 while 循环中条件表示方式的考察。当 while 后跟整数的情况下，当数字为 0 时，不会进入循环，否则表示无限循环。当输入的 n 为负整数时，进入循环，变量 n 每次减 1，会越来越小，会进入无限循环，所以本题错误。

7. 执行以下 Python 代码，将输出 π 的值。

```
import math
print(math.pi())
```

【答案】 错误

【解析】 本题考查数学库 math 的相关知识点。 π 是数学库 math 中的一个常量，用 math.pi 表示，不用加括号，所以本题错误。

8. Python 是一种低级语言，适合初学者程序设计入门。

【答案】 错误

【解析】 本题属于对程序设计语言的考察。Python 属于高级语言之一，所以本题错误。

9. {1,1}在 Python 中是合法的表达式。

【答案】正确

【解析】本题是对 Python 下常见数据类型的考察。 $\{1, 1\}$ 是集合 set 的表示形式，不过集合是由不重复的元素组成的，如果定义一个变量 $a=\{1, 1\}$ ，定义完成后 $a=\{1\}$ 。

10. 以下 Python 代码执行时将报错，因为 a 初始被赋值为整数，随后将其赋值为字符串，故错误。

```
a = 123
a = "ABC"
```

【答案】错误

【解析】本题主要是对变量赋值的考察。第一次赋值，变量 $a=123$ ，第二次赋值，变量 $a="ABC"$ ，第二次赋值会覆盖第一次的值，运行时不会报错，所以本题错误。

三、编程题

第 1 题

【问题描述】

小明刚刚学习了素数的概念：如果一个大于 1 的正整数，除了 1 和它自身外，不能被其他正整数整除，则这个正整数是素数。现在，小明想找到两个正整数 A 和 B 之间（包括 A 和 B）有多少个素数。

【输入描述】

输入 2 行，第一行包含正整数 A，第二行包含正整数 B。约定 $2 \leq A \leq B \leq 1000$ 。

【输出描述】

输出一行，包含一个整数 C，表示找到 C 个素数。

【样例输入 1】

2
10

【样例输出 1】

4

【样例解释 1】

在 2 和 10 之间有 4 个素数，分别为：2、3、5、7。

【样例输入 2】

98

100

【样例输出 2】

0

【题目大意】

求素数个数问题，输入两个数字，依次判断这两个数字之间的每一个数是否为素数，即找到除了 1 和它自身外，不能被其它正整数整除的数字。

【解题思路】

本题主要考察多层 for 循环的嵌套使用。

1. 首先输入两个数字，借助 for 循环，表示出这两个数字之间的每一个数。
2. 每取到一个数，判断这个数字是否为素数，再借助一个 for 循环，表示出判断的范围，要找的是除了 1 和它本身之外的，如果有能够整除的，说明不是素数，否则是素数。
3. 设置变量，记录素数的个数。
4. 按照要求完成输出。

【参考程序】

```
n = int(input()) # 输入第一个数
m = int(input()) # 输入第二个数
tnt = 0 # 创建变量，记录素数个数
for i in range(n,m+1): # i 表示从 n 到 m 之间的每一个数字
    for j in range(2,i): # j 的范围是从 2 到 i-1
        if i % j == 0: # 如果能够整除，表示不是素数，终止循环
            break
    else: # 从 2 到 i-1 之间的数字没有能够整除的，执行这部分
        tnt +=1 # 素数个数加 1
print(tnt)
```

第 2 题

【问题描述】

1. 判断一个正整数是否为自幂数；
2. 自幂数是指 N 位数各位数字 N 次方之和是本身，如 153 是 3 位数，其每位数的 3 次方之和是 153 本身，因此 153 是自幂数，1634 是 4 位数，其每位数的 4 次方之和是 1634 本身；

3. 首先输入一个整数 M 表示随后有 M 行每行有一个整数，输出每个整数是否为自幂数，如是则输出 T，否则 F。

【输入描述】

1. 输入正整数 M ($1 \leq M \leq 100$)。不考虑输入不合规情形如负数、负整数、零或其他字符串等；
2. 从第 2 行开始，输入 M 行每行一个正整数($1 \leq M \leq 2147483647$)；
3. **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. 输出对应正整数是否为自幂数，如是则输出一个英文字母 T，否则一个英文字母 F；
2. **特别提示：** 注意输出字母为大写，小写将判为错误，前后没有空格等。

【样例输入 1】

3
152
111
153

【样例输出 1】

F
F
T

【样例输入 2】

5
8208
548834
88593477
12345
5432

【样例输出 2】

T
T
T
F
F

【题目大意】

查找自幂数个数问题，输入 M 个数字，依次判断是否是自幂数

【解题思路】

本题主要考察输入固定个数字以及对于一个数字依次获取各位上的数的内容。

1. 首先输入数据的数量 N
2. 执行 N 次，每次输入一个数字，求出这个数字各位幂次方的和。
3. 判断是否是自幂数，按照要求完成输出。

【参考程序】

```
#自幂数
N = int(input()) #输入数据的数量
for i in range(N):
    M = int(input()) #输入判断的数
    howDigit = 0
    #获得输入整数的位数
    tmp = M
    while tmp != 0:
        howDigit += 1
        tmp //= 10 #执行完本步，M 将减少 1 位
    tnt = 0
    tmp = M
    #计算各位幂次方的和
    while tmp != 0:
        tnt = tnt+ ( tmp % 10) ** howDigit
        tmp //= 10
    #判断是否为自幂数
    if tnt == M:
        print("T")
    else:
        print("F")
```