

2023 年 GESP9 月认证 Python 三级试卷解析

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 三级认证真题解析。

一、单选题（每题 2 分，共 30 分）

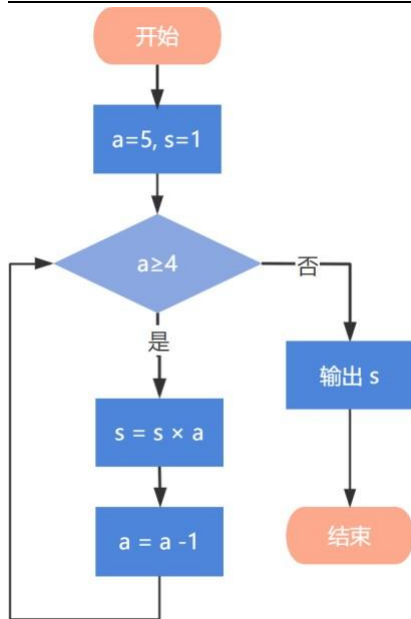
1. 人们所使用的手机上安装的 App 通常指的是（ ）。

- A. 一款操作系统
- B. 一款应用软件
- C. 一种通话设备
- D. 以上都不对

【答案】B

【解析】本题属于考察计算机基础知识。APP 是 Application 的缩写，通常指的是应用程序。在计算机领域，APP 是指一种可以在特定操作系统上运行的软件程序，它为用户提供了特定的功能和服务。例如，手机上的微信、支付宝等应用就是 APP。故正确答案为 B 选项。

2. 下列流程图的输出结果是（ ）。



- A. 60
- B. 20
- C. 5
- D. 1

【答案】 B

【解析】 本题属于考察流程图的知识。该流程图主要进行条件判断，当 $a \geq 4$ 时，执行 $s = s \times a$ ， $a = a - 1$ 。否则，输出 s 。该流程图执行两次，分别是 $a = 5$ 和 $a = 4$ ，所以 $s = 1 \times 5 \times 4$ 。故正确答案为 B 选项。

3. 已知大写字母'A'的 ASCII 编码的十六进制表示为 0x41，则字符'L'的 ASCII 编码的十六进制表示为 ()。

- A. 4A
- B. 4B
- C. 4C
- D. 52

【答案】 C

【解析】 本题属于考察进制转换的知识。十六进制是一种基数为 16 的计数系统，它使用 16 个不同的符号来表示数值，包括 0 到 9 的数字和 A 到 F 的字母。在 Python 中使用一个以 0x 开头的数字时，Python 会自动将其视为十六进制数。

当 A 为 41 时，按照 26 个英文字母顺序，L 就是 4C。故正确答案为 C 选项。

4.下面有关 Python 的 dict 类型 key（键）的说法，正确的是（ ）。

- A. dict 类型的 key 可以是 list 数据
- B. dict 类型的 key 可以是 set 数据
- C. dict 类型的 key 可以是 dict 数据
- D. dict 类型的 key 可以是 tuple 数据

【答案】D

【解析】本题属于考察 Python 字典键值对基本格式的知识。在 Python 中，字典（dict）是一种可变的、无序的、键值对（key-value）集合。字典中的键（key）是唯一的，用于标识字典中存储的值（value）。字典的键可以是任何不可变的数据类型，如字符串、数字或元组。故正确答案为 D 选项。

5.下面 Python 赋值语句，不正确的是（ ）。

- A. a = [12, 12]
- B. b = (12, 12)
- C. c = {12, 12}
- D. 以上没有不正确

【答案】D

【解析】本题属于考察 Python 赋值和列表、集合、元组的基本应用的知识。在 Python 中，集合是一个无序的、不重复的元素序列。当我们创建一个集合时，如果其中存在重复的数据，那么这些重复的数据会被自动去重，只保留一个。程序不会报错。A 选项为列表，B 选项为元组，C 选项为集合。故正确答案为 D 选项。

6.有关下面 Python 代码的描述，正确的是（ ）。



```
a = (1, 2, 3, 4)
b = (5, 6, 7, 8)
a = a[1:] + b[:1]
print(a)
```

- A. 上述代码执行将报错
- B. 加号运算符不适用 tuple 类型
- C. tuple 是不可更改类型，不能切片（slice）操作
- D. 输出结果是(2, 3, 4, 5)

【答案】D

【解析】本题属于考察 Python 元组的基本应用的知识。在 Python 中，元组(tuple)是 Python 中的一种不可变序列类型。可以使用索引访问元组中的元素，索引从 0 开始，可以结合切片操作获取元组中的子序列。切片操作的两个操作数均为可选。如果缺少第一个操作数，则从起始处开始切片。如果缺少第二个操作数，则切片持续到末尾。切片操作的起始索引是包含的，而结束索引是不包含的。例如，my_tuple[1:4]表示从索引 1 开始，到索引 4 之前的所有元素。故正确答案为 D 选项。

7. 下列 Python 赋值语句错误的是 ()。

- A. a = [1, 2] + [3, 4]
- B. b = (1, 2) + (3, 4)
- C. c = {1, 2} + {3, 4}
- D. d = "1, 2" + "3, 4"

【答案】C

【解析】本题属于考察 Python 列表、集合、元组和字符串拼接的知识。在 Python 中，集合是一个无序的、不重复的元素序列，通过 set()函数创建。按 set 特征，重复会自动清除保留一个。使用时会抛出 TypeError 异常。如果需要合并两个集合，可以使用集合的 update()函数或者 union()函数来实现。故正确答案为 C 选项。

8.下面 Python 代码执行后的输出是 ()。

```
lstA = [11, 12, 13, 14, 15, 16, 17]
print(lstA[3: 2], end = ",")
print(lstA[-5: -3])
```

- A. [],[13, 14]
- B. [13, 14],[]
- C. [13, 14],[13, 14]
- D. [],[]

【答案】 A

【解析】本题属于考察 Python 列表切片的知识。列表切片是 Python 中对列表进行部分提取的一种函数，它可以让我们从列表中获取一部分元素。切片操作的语法是 `list[start:end]`，其中 `start` 表示起始索引，`end` 表示结束索引（`end` 不包含在切片结果中）。切片操作会返回一个新的列表，包含从 `start` 到 `end-1` 的元素。如果省略 `start`，则默认从列表开头开始；如果省略 `end`，则默认切到列表末尾。索引可以是负数，表示从列表末尾开始计数。例如，`my_list[-1]`表示获取列表的最后一个元素。如果切片操作中的 `start` 大于 `end`，或者省略了 `start` 和 `end`，则会返回一个空列表。例如，`my_list[:0]`或 `my_list[::]`都会返回一个空列表。故正确答案为 A 选项。

9.下面 Python 代码执行后的输出是 ()。

```
strA = "A1x"
strA = sorted(strA)
print(strA)
```

- A. ['1', 'A', 'x']
- B. ['1', 'x', 'A']
- C. "1Ax"
- D. "A1x"

【答案】A

【解析】本题属于考察Python内置函数sorted()函数的知识。sorted()是Python中的一个内置函数，用于对可迭代对象进行排序。它会返回一个新的已排序的列表，而不会修改原始的可迭代对象。默认情况下，sorted()函数会按照元素的升序进行排序。如果需要降序排序，可以设置参数reverse=True。字符串的排序顺序是按照字符的Unicode编码值进行排序的。对于ASCII码范围内的字符（即0-127），其排序顺序与ASCII码表的顺序相同。在ASCII中，数字小于大写字母，大写字母小于小写字母。1，A，x的ASCII码分别为49，65，120。故正确答案为A选项。

10.下面Python代码执行后的输出是（ ）。

```
lstA = [2, -3, 4, 3, 5, 6, 7, 8, -2]
lstA.sort(key = abs, reverse = True)
print(lstA)
```

- A.[8, 7, 6, 5, 4, -3, 3, -2, 2]
- B. [8, 7, 6, 5, 4, -3, 3, 2, -2]
- C. [2, -2, -3, 3, 4, 5, 6, 7, 8]
- D. [-2, 2, -3, 3, 4, 5, 6, 7, 8]

【答案】B

【解析】本题属于考察Python内置函数sort()函数和abs()函数的知识。sort()函数是Python中的一个内置函数。默认情况下，sort()函数会按照元素的升序进行排序。sort函数还有两个可选参数：key和reverse，key在使用时必须提供一个排序过程总调用的函数，如果需要降序排序，可以设置参数reverse=True。abs()是Python中的一个内置函数，用于计算数字的绝对值。它接受一个数字作为参数，并返回该数字的绝对值。所以本题是降序排列的，当遇到相同数据时，sort()函数排序是稳定排序，即如果排序值相同，则维持原顺序不变，在前仍在前，会根据元素从左到右出现顺序进行排序。故正确答案为B选项。

11.下面Python代码执行后的输出是（ ）。

```
setA = set("Python")
setB = set("python")
print(setA - setB)
```

- A. 'P'
- B. 'p'
- C. {'P'}
- D. {'p'}

【答案】C

【解析】本题属于考察 Python 集合基本应用差集的知识。在 Python 中，可以使用“-”运算符来计算集合的差集。所有属于 A 且不属于 B 的元素构成的集合，叫做集合 A 与集合 B 的差集。故正确答案为 C 选项。

12.某机构职工编码规则是“入职年份顺序码”，其中入职年份为 2 位数，顺序码 3 位数表示入职顺序。下面代码要求输入职工编码，并在输入大写字母 Q 时结束输入，要求输出每个年份入职人数，应在横线填入代码是（ ）。

```
statData = {} #保存每个年份对应的人数
while True:
    ID = input("请输入职工编号: ")
    if ID == "Q":
        break
    Year = ID[:2] #得到入职年份
    _____
print(statData)
```

- A. statData[Year] += statData.get(Year) + 1
- B. statData[Year] = statData.get(Year, 0) + 1
- C. statData[Year] = statData.get(Year, None) + 1
- D. statData[Year] += 1

【答案】B

【解析】本题属于考察 Python 字典中 get () 函数和字符串切片的知识。字典 get() 函数是 Python 中的一个内置函数，用于从字典中获取指定键对应的值。如果键不存在于字典中，那么返回一个默认值。get() 接受两个参数，第一个参数是要查找的键，第二个参数是可选的默认值，当键不存在时返回该默认值。题目要求入职年份为 2 位数，顺序码 3 位数表示入职顺序，形如“23001”的格式。输出

每个年份入职人数，所以正确答案为 B 选项。

13. Python 表达式 `[x for x in range(2,20) if x % 3 and x % 5 == 0]` 的值是 ()。

- A. [15]
- B. [5, 10]
- C. [2, 4, 7, 8, 11, 13, 14, 16, 17, 19]
- D. [3, 6, 9, 12, 18]

【答案】 B

【解析】 本题属于考察 Python 列表推导式的基本应用。该推导式是得到 2 到 19 中不能被 3 整除，但是能被 5 整除的数组成的列表。故正确答案为 B 选项。

14. 下面 Python 代码用于实现十进制到 M 进制 (M 为二进制到十六进制) 的转换，横线处应填入代码是 ()。

```
N = int(input("请输入正整数: "))
M = int(input("请输入转换进制, 不能大于 16: "))
Alpha="0123456789ABCDEF"

rst = "" #保存转换结果
while N != 0:
    Remainder = N % M #Remainder表示余数
    _____
    N = N // M
print(rst)
```

- A. `rst += Alpha[Remainder]`
- B. `rst = Alpha[Remainder] + rst`
- C. `rst = rst + Alpha[Remainder]`
- D. `rst = str(Remainder) + rst`

【答案】 B

【解析】 本题属于考察 Python 进制转换的知识。十进制转 N 进制的规则是将

一个十进制数采用辗转相除法除以 N，然后将余数倒序排列。这个过程一直持续到商为 0 为止。最后得到的余数序列就是对应的二进制表示。因为是倒叙排列，所以是先得到当前余数对应的字符，再拼接之前得到的字符和 rst，故正确答案为 B 选项。

15.下面 Python 代码实现字符串乱序【随机打乱】，横线处应填入代码（ ）。

```
import random
a = "123ABCxyz"
a = list(a)
a.sort(_____)
print("".join(a))
```

- A. key = random.random
- B. key = lambda x:random.random(x)
- C. key = lambda x:random.random()
- D. key = lambda x:random.random

【答案】C

【解析】本题属于考察 Python 内置函数 sort()函数，匿名函数 lambda，list()函数和 random()函数知识的结合使用。lambda 函数的语法如下：lambda 参数列表：表达式。sort()函数会按照元素的升序进行排序。sort()函数还有两个可选参数：key 和 reverse，key 在使用时必须提供一个排序过程总调用的函数。random()函数是 Python 中的一个内置函数，用于生成一个 0 到 1 之间的随机浮点数，无参数。每次得到不一样的结果。故正确答案为 C 选项。

二、判断题（每题 2 分，共 20 分）

1. 我们常说的互联网（Internet）是一个覆盖全球的广域网络，它不属于任何一个国家。

【答案】正确√

【解析】本题考察计算机网络的基本知识。互联网，又称国际网络，指的是网络与网络之间所串连成的庞大网络，这些网络以一组通用的协议相连，形成逻辑上

的单一巨大国际网络。互联网是全球性的。这就意味着这个网络不管是谁发明了它，是属于全人类的。所以本题正确。

2. 二进制数 101.101 在十进制下是 5.005。

【答案】错误×

【解析】本题考察进制转换的知识。整数的二进制转化为十进制数，从二进制数的右边第一位起，从右往左，先用二进制位置上的数乘以 2 的相应位数的幂，然后把每一位的乘积相加即可得到二进制数对应的十进制数。小数二进制转化为十进制数，从小数点后第一位上的二进制数字乘以 2 的负一次方加上第二位上的二进制数字乘以 2 的负二次方，以此类推第 n 位上的二进制数字乘以 2 的负 n 次方。 $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 5.625$ 。所以本题错误。

3. Python 内置函数 len()对 list、tuple 和 str 有效，对 set 和 dict 无效。

【答案】错误×

【解析】本题考察 Python 内置函数 len()函数的基本知识。len()函数是 Python 中的一个内置函数，用于获取对象的长度。它可以对以下类型的对象进行操作：1.字符串 (str)：返回字符串中字符的个数。2.列表 (list)：返回列表中元素的个数。3.元组 (tuple)：返回元组中元素的个数。4.字典 (dict)：返回字典中键值对的个数。5.集合 (set)：返回集合中元素的个数。所以本题错误。

4. Python 表达式 len(set("12,34,56"))的值为 3。

【答案】错误×

【解析】本题考察 Python 内置函数 set()函数和 len()函数的基本知识。set()函数用于创建一个无序且不重复的元素集合。当我们将字符串放入 set()函数时，它会将字符串中的每个字符视为独立的元素，并将它们添加到集合中。所以是{'1', '2', ',', '3', '4', '5', '6'}。len()函数是 Python 中的一个内置函数，用于获取对象的长度。最后值为 7。所以本题错误。

5. Python 表达式 $2\&3$ 的值为 5.

【答案】错误×

【解析】本题考察 Python 位运算与 ($\&$) 的知识。 $\&$ (按位与) 对应二进制下的 AND 操作, 如果两个相应的二进制位都为 1, 则该位的结果为 1, 否则为 0。2 的二进制是 0010, 3 的二进制是 0011。0010 $\&$ 0011 结果是 0010, 转为十进制是 2。所以本题错误。

6. 执行 Python 代码 `print("李白".replace("白","杜"))`后将输出"李杜"。

【答案】正确√

【解析】本题考察 Python 内置函数 `print()` 函数和 `replace()` 函数的基本知识。在 Python 中, `replace()` 是一个字符串方法, 用于将字符串中的某个子串替换为另一个子串。它接受两个参数, 第一个参数是要被替换的子串, 第二个参数是替换后的子串。所以本题正确。

7. 执行 Python 代码 `print(list(filter(lambda n:n%5,range(10))))`后将输出[0, 5]。

【答案】错误×

【解析】本题考察 Python 内置函数 `list()` 函数、`filter()` 函数、`range()` 函数和匿名函数 `lambda` 的基本知识。`filter()` 函数接受一个匿名函数和一个可迭代对象 `range(10)` 作为参数, 并生成一个满足条件对 5 取余不为 0 的元素组成的迭代器。然后使用 `list()` 函数返回由符合条件元素组成的新列表。结果为 [1,2,3,4,6,7,8,9]。所以本题错误。

8. 执行 Python 代码 `print([i for i in range(10) if i % 5 == 0])`后将输出[0, 5]。

【答案】正确√

【解析】本题考察 Python 列表推导式的知识。本题输出 0 到 9 之间能被 5 整除的数字, 所以本题正确。

9. 下面 Python 代码执行时输入 10,20 后将输出 30。

```
a,b = input().split(",")
print(a + b)
```

【答案】 错误×

【解析】 本题考察 Python 内置函数 `input()` 函数，`split()` 函数和列表拼接的基本知识。`split()` 是 Python 中的一个字符串方法，结合使用用于将一个字符串按照指定的分隔符进行输入多个数据。`input()` 函数返回值是 `string` 类型，输出为 `1020`。所以本题错误。

10. 在 Python 代码中先执行 `a = (5,6,3,10)`，然后执行 `print(a.sort())` 其输出结果是 `None`。

【答案】 错误×

【解析】 本题考察 Python 元组和内置函数 `sort()` 函数的知识。元组不能使用 `sort()` 函数，`sort()` 函数只能对列表进行排序，不能对元组、集合等其他类型的序列进行排序。如果需要对其他类型的序列进行排序，可以先将其转换为列表，然后使用 `sort()` 函数进行排序。所以本题错误。

三、编程题（每题 25 分，共 50 分）

1. 小杨的储蓄

【问题描述】

小杨共有 N 个储蓄罐，编号从 0 到 $N-1$ 。从第 1 天开始，小杨每天都会往存钱罐里存钱。具体来说，第 i 天他会挑选一个存钱罐 a_i ，并存入 i 元钱。过了 D 天后，他已经忘记每个储蓄罐里都存了多少钱了，你能帮帮他吗？

【输入描述】

输入 2 行，第一行两个整数 N, D ；第二行 D 个整数，其中第 i 个整数为 a_i （保证 $0 \leq a_i \leq N - 1$ ）每行的各个整数之间用单个空格分隔。保证 $1 \leq N \leq 1,000$ ； $1 \leq D \leq 1,000$ 。

【输出描述】

输出 N 个用单个空格隔开的整数，其中第 i 个整数表示编号为 $i-1$ 的存钱罐中有多少钱（ $i = 1, \dots, N$ ）。

【样例输入 1】

2 3

0 1 0

【样例输出 1】

4 2

【样例输入 2】

3 5

0 0 0 2 0

【样例输出 2】

11 0 4

【题目大意】

小杨有 N 个存钱罐，存了 D 天（第一行输入数据），每天会选择其中一个存钱罐（第二行输入的数据就是 D 个数据，第 i 个数据就是第 i 天选择的是第几个存钱罐）存入第 i 天的 i 元钱。问最终 N 个存钱罐，每个存钱罐里面有多少元（输出的 N 个数据）。

【解题思路】

本题主要考察内置函数和列表推导式的知识。

1. 输入两个数据 n, d ，单个空格分隔，使用`split()`函数，使用`map()`函数将其转换为整数。
2. 输入 D 个数据，通过`map()`函数将每个数据转换为`int`类型，并使用`list()`函数转为列表，赋值给 x 。
3. 使用一个列表推导式，创建一个包含 n 个元素的列表 arr ，存储存钱罐对应的总钱数。其中每个元素都初始化为 0 。
4. 遍历 d ，将对应的 i 作为 x 列表的索引值，找到对应的天数对应的罐子编号，将编号传入 arr 列表，作为索引值。将 $i+1$ 存入对应的位置（因为`range()`函数是从 0 开始遍历，所以是 $i+1$ ）。

5.将 `arr` 列表中的每个元素通过 `map()`函数转为 `str` 类型,使用 `join()`函数转为以空格连接的字符串。

【参考程序】

```
n, d = map(int,input().split(' '))
x = list(map(int, input().split(' ')))
arr = [0 for i in range(n)]
for i in range(d):
    arr[x[i]] += (i + 1)
print(' '.join(map(str, arr)))
```

2. 进制判断

【问题描述】

进制数指的是逢 N 进一的计数制。例如,人们日常生活中大多使用十进制计数,而计算机底层则一般使用二进制。除此之外,八进制和十六进制在一些场合也是常用的计数制(十六进制中,一般使用字母 `A` 至 `F` 表示十至十五)。

现在有 N 个数,请你分别判断他们是否可能是二进制、八进制、十进制、十六进制。例如, `15A6F` 就只可能是十六进制,而 `1011` 则是四种进制皆有可能。

【输入描述】

输入的第一行为一个十进制表示的整数 N 。接下来 N 行,每行一个字符串,表示需要判断的数。保证所有字符串均由数字和大写字母组成,且不以 `0` 开头。保证不会出现空行。

保证 $1 \leq N \leq 1000$, 保证所有字符串长度不超过 10。

【输出描述】

输出 N 行,每行 4 个数,用空格隔开,分别表示给定的字符串是否可能表示一个二进制数、八进制数、十进制数、十六进制数。使用 1 表示可能,使用 0 表示不可能。

例如,对于只可能是十六进制数的 `15A6F`,就需要输出 `0001`;而对于四者皆有可能的 `1011`,则需要输出 `1111`。

【样例输入 1】

2

15A6F

1011

【样例输出 1】

0 0 0 1

1 1 1 1

【样例输入 2】

4

1234567

12345678

FF

GG

【样例输出 2】

0 1 1 1

0 0 1 1

0 0 0 1

0 0 0 0

【题目大意】

先输入数字 N ，表示接下来要输入 N 行数据，判断这些数据是否可能表示一个二进制数、八进制数、十进制数、十六进制数。使用 1 表示可能，使用 0 表示不可能。输出 N 行判断结果。

【解题思路】

本题主要考察内置函数,列表和字符串函数的知识。

1. 通过 `input()`输入 n ,使用 `int()`函数转为 `int` 类型。
2. 通过 `for...in...`循环去实现输入 n 个数据，命名为 `st`。
3. 使用 `strip()`函数，得到具体数据。
4. 设置临时最大值 `max_bit`,赋值为‘0’。
5. 遍历 `st`，将 `st` 字符中的最大字符赋值给 `max_bit`。



6. 设置一个 `answer` 列表，列表项分别判断是否大于对应进制的最大数值。最后输出判断结果。

【参考程序】

```
n = int(input())
for i in range(n):
    st = input().strip()
    max_bit = '0'
    for ch in st:
        if ch > max_bit:
            max_bit = ch
    answer = [
        max_bit <= '1',
        max_bit <= '7',
        max_bit <= '9',
        max_bit <= 'F',
    ]
    print(' '.join(map(str,map(int, answer))))
```