

2023 年 6 月 GESP 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 6 月份 Python 三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	D	D	C	D	D	C	A	B	A	D	C	B

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. 二进制数 11.01 在十进制下是（ ）。

- A. 3.01
- B. 3.05

- C. 3.125
- D. 3.25

【答案】D

【解析】本题主要考查二进制转十进制的内容。二进制转十进制时，可以采用按权展开、逐项相加的方法，小数点之前，分别是2的0次方，2的1次方，2的2次方...，小数点之后，依次是乘以2的负一次方，2的负二次方，2的负三次方...，那么 $11.01 = 1 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2} = 3.25$ ，所以本题选择D选项。

3. 下面有关 Python 列表的描述，正确的是（ ）。
- A. Python 列表的每个成员必须是相同的数据类型
 - B. 一旦确定 Python 列表的成员，不可新增成员
 - C. 不可删除 Python 列表的成员
 - D. 以上说法都不正确

【答案】D

【解析】本题考察考生对数据类型列表的了解，列表是一可变数据类型，可以增加、删除和修改等，并且列表里的元素可以是整数、小数、字符串、列表和元组等类型，综合考虑，本题选择D选项。

4. 下面 Python 赋值语句，不合法的是（ ）。
- A. `a,b,c = [1, 2, 3]`
 - B. `b = 2, 3, 4`
 - C. `d = (3,)`
 - D. 以上都合法

【答案】D

【解析】本题主要考察列表和元组的定义。A选项中是一个多重赋值语句，表示将列表[1, 2, 3]中的元素分别赋值给变量a、b和c；B选项和C选项都是元组，需要注意的是，定义元组只有一个元素时，也要加逗号；所以本题中几个选项都合法，选择D选项。

5. 有关下面 Python 代码的描述，正确的是（ ）。

```
a = (1, 2)
a = a + (3, 4)
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，因此两个 tuple 不可相加。
- D. 输出结果是(1, 2, 3, 4)

【答案】D

【解析】本题主要考察元组的合并。元组是不可变的数据类型，但是元组之间支持加号运算符，元组+元组表示元组的合并，首先变量 a 等于元组(1, 2)，然后把元组 a 和(3, 4)合并的内容赋值给 a，最后变量 a=(1, 2, 3, 4)。所以本题选择 D 选项。

6. Python 表达式 `len(set("长城, 颐和园"))-len(set(["长城", "颐和园"]))` 的值是 ()。
- A. 0
 - B. 2
 - C. 4
 - D. 6

【答案】C

【解析】本题主要考察字符串、列表和集合之间的转换。`set("长城, 颐和园")` 的结果是 {'园', '长', '城', '和', '颐', ','}, 而 `set(["长城", "颐和园"])` 的结果是 {'长城', '颐和园'}, 通过 `len()` 获取元素个数再相减，所以本题选择 C 选项。

7. 下面 Python 代码执行后的输出是 ()。

```
a = [1,2,3,4,5,6]
for i in a:
    if i % 2 == 1:
        del i
print(a)
```

- A. [1, 3, 5]
- B. [2, 4, 6]
- C. []
- D. 执行过程将报错

【答案】D

【解析】本题主要考查遍历列表过程中删除元素的特点。首先通过 `len()` 获取列表里元素个数为 6，通过 `for` 循环遍历 `i` 的范围是 0 到 5，满足 `i%2==1` 的有 1、3、5 三个数字，首先删除的是 `a[1]`，即元素 2 被删掉，然后后面的元素会前移，此时，列表里已经没有 `a[5]` 这个元素，后面再删除就会报错，所以本题选择 D 选项。

8. 下面 Python 代码执行后的输出是（ ）。

- A. [1, 3, 5]
- B. [2, 4, 6]
- C. []
- D. [1, 2, 3, 4, 5, 6]

【答案】D

【解析】本题主要考察的是 `del` 方法，`del 列表[索引]` 表示删除列表里指定索引的元素，`del 列表` 表示删除整个列表。这道题目中，`del` 后删除的是一个数字 `i`，跟列表 `a` 是没有关系的，所以列表 `a` 不变，本题选择 D 选项。“`del 变量`”表示删除变量，此处可以删除变量 `i`，但每次循环时将重新将 `a` 的一个成员值赋值给 `i`。删除 `i` 值不影响 `a` 中成员的值，循环时是赋值 `a` 中成员的值到 `i`，而不是直接删除 `a` 中成员。

9. 下面 Python 代码执行后的输出是（ ）。

```
a = [1,2,3,4,5,6]
a.sort(key = lambda x:x%2)
print(a)
```

- A. [1, 2, 3, 4, 5, 6]
- B. [1, 3, 5, 2, 4, 6]
- C. [2, 4, 6, 1, 3, 5]
- D. [0, 0, 0, 1, 1, 1]

【答案】 C

【解析】 本题主要考察列表.sort()函数和lambda函数的用法。列表.sort()表示直接对列表元素进行排序，当指定key参数时，列表.sort()函数会使用该参数指定的函数对列表中的每个元素进行处理，并根据处理后的结果进行排序。排序将根据处理后的结果来确定元素的顺序，而不是直接比较原始的元素本身。而lambda函数表示创建匿名函数，没有函数名，只有参数列表和函数体。

这道题目中，对列表a中的每个元素对2取余，然后根据区域的结果进行排序，所以本题选择C选项。要注意：如果有key参数，则排序按key所指定的函数处理每一个成员后的值排序作为排序依据。不修改原有成员的值。

10. 下面Python代码执行后的输出是（ ）。

```
set1 = {2, 4, 6, 8}
set2 = {8, 6, 2, 4, 8}
print(set1 == set2)
```

- A. True
- B. False
- C. 1
- D. 0

【答案】 A

【解析】本题主要考察集合的特点和集合的比较。集合中的元素是无序且不重复的。因此，无论元素的顺序如何，只要两个集合中的元素相同，它们就被认为是相等的。集合 set1 中的元素有 2、4、6、8，集合 set2 中去掉重复的数字是 8、6、2、4，两个集合的元素相同，所以比较相等的结果为 True。所以本题选择 A 选项。

11. 下面 Python 代码执行后的输出是（ ）。

```
tplA=(1,4,7,2,5,8)
print(list(reversed(tplA)))
```

- A. (8, 5, 2, 7, 4, 1)
- B. [8, 5, 2, 7, 4, 1]
- C. [8, 7, 5, 4, 2, 1]
- D. (8, 7, 5, 4, 2, 1)

【答案】B

【解析】本题主要考查 reversed() 反转函数和 list() 列表转换函数。首先把元组里的元素进行翻转，然后通过 list() 函数转换为列表，结果为 [8, 5, 2, 7, 4, 1]，所以本题选择 B 选项。reversed() 只是翻转，不排序。

12. Python 语句 print([str(n*n+1)*(n*n+1) for n in range(3)]) 执行后的输出是（ ）。

- A. ['1', '22', '55555']
- B. ['22', '55555', '10101010101010101010']
- C. ['22', '55555']
- D. 执行将报错

【答案】A

【解析】本题主要考察列表推导式的内容。本题使用列表推导式生成一个列表，其中的元素是对每个数字 n 进行计算的结果。在这个例子中，数字 n 的取值范围是 range(3)，即 0、1 和 2。对于每个 n，表达式 str(n*n+1)*(n*n+1) 表示把 (n*n+1) 转换为字符串后，重复该字符串 (n*n+1) 次。比如 n=0 时，(n*n+1) 等于 1，执行 str(n*n+1)*(n*n+1) 后为 '1'。所以本题选择 A 选项。

13. 下面 Python 代码执行后的输出是（ ）。

```
b = sorted("Python is a popular  
programming language".split(),  
key=str.lower)  
print(b[-1])
```

- A. a
- B. y
- C. programming
- D. Python

【答案】 D

【解析】 本题主要考察字符串的几个函数的使用和字符串的大小比较。字符串 `.split()` 表示拆分，括号里不写参数时默认以空格拆分，返回值是一个列表；字符串 `.lower()` 表示把字母转为小写；`sorted()` 表示排序，会生成一个新的列表，指定 `key` 参数时，`sorted()` 函数会使用该参数指定的函数对排序对象中的每个元素进行处理，并根据处理后的结果进行排序。

这题中首先使用 `split()` 方法进行拆分，然后通过指定的 `key` 参数 `str.lower` 把拆分后的列表里元素的字母转为小写，再进行排序，字符串的排序会根据 ASCII 值的大小来排，最终排序结果为

`['a', 'is', 'language', 'popular', 'programming', 'Python']`，所以本题选择 D 选项。

14. 下面 Python 代码执行后的输出是（ ）。

```
groveA =  
['Cherry', 'Apple', 'Banana', 'Apple']  
groveB = ['Apple', 'Peach', 'Peach']  
Fruits = set(groveA + groveB)  
n = len(Fruits)  
print(n)
```

- A. 7

- B. 5
- C. 4
- D. 执行将报错

【答案】C

【解析】本题主要考查列表的拼接以及列表和集合的转换内容。groveA+groveB表示把两个列表合并起来，然后通过 set() 转换成集合，集合是由不重复的元素组成的，所以会把列表里重复的元素去掉，再统计个数是 4，所以本题选择 C 选项。

15. 已知大写字符 X 的 ASCII 编码的 16 进制表示为 58₁₆，则字符 Q 的十六进制 ASCII 编码是 ()。
- A. 47
 - B. 51
 - C. 53
 - D. 54

【答案】B

【解析】本题考查 ASCII 码的 16 进制表示方法。大写字符 X 和大写字符 Q 之间间隔 7 位，所以在 58 的基础上直接减掉 7，所以本题选择 B 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

1. 一个算法可以用不同的形式来描述，但要求描述比较规范，因此不能用自然语言描述。

【答案】错误

【解析】本题主要考察算法的描述形式。一个算法可以用不同的形式来描述，包括自然语言、伪代码、流程图等等。自然语言是一种常见的算法描述方式，尽管它可能不如形式化的描述方式那样精确和规范，但在某些情况下仍然可以有效地传达算法的思想和步骤。所以本题错误。

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. 执行 Python 语句 `a = {}` 后的变量的 `a` 类型可能是 `set` 也可能是 `dict`。

【答案】错误

【解析】本题主要考察字典和集合的创建。在 Python 中，使用大括号 `{}` 创建的被默认解释为字典 (`dict`)。要创建一个空的集合，需要使用 `set()` 函数。所以本题错误。

4. Python 表达式 `len(set("CCF"))` 的值为 2。

【答案】正确

【解析】本题主要考察集合的创建和特点。集合是无序且元素唯一的，重复的字符被去除，所以 `set("CCF")` 的结果是 `{'F', 'C'}`，再通过 `len()` 获取元素个数为 2，所以本题正确。

5. 下面 Python 代码执行后的输出是 Python is a popular programming language.

```
a = "Python is a popular programming
language."
a.replace("programming", "computer")
print(a)
```

【答案】正确

【解析】本题主要考察字符串的 `replace()` 替换函数。字符串的 `replace()` 函数不会修改原始字符串，而是返回一个新的修改后的字符串。所以这题中我们替换完成后打印，打印的是原字符串，所以本题正确。

6. Python 的 `dict` 类型，其键 (`key`) 可以是 `tuple` 类型，但不可以是 `list` 类型，因此下列代码执行不会报错。

```
tab99 = {}
for i in range(1, 9 + 1):
    for j in range(1, i + 1):
        tab99[(j, i)] = i*j
print(tab99)
```

【答案】 正确

【解析】 本题主要考察字典的内容。字典的键具有唯一性，并且是不可变的类型，列表和字典不可以作为字典的键，字符串、数字和元组都可以作为字典的键。所以本题正确。

7. 下面 Python 代码执行后将输出[1, 3, 5]。

```
a = [1,2,3,4,5,6]
print(filter(lambda x:x%2,a))
```

【答案】 错误

【解析】 filter() 函数执行后将得到一个 filter() 类型，支持 for-in 循环。如果要输出 1, 3, 5 需要修改为 print(list(filter(lambda x:x%2, a)))

8. 下面 Python 执行后的输出是 True。

```
a, b = "12,2".split(",")
if a>b:
    print(True)
else:
    print(False)
```

【答案】 错误

【解析】 本题主要考查字符串拆分的特点。字符串.split(‘分割符’) 以指定的分割符来对字符串拆分，不写时默认以空格来拆分，拆分的结果是一个列表。需要注意的是，列表里的每一个元素都是字符串类型，所以本题中以逗号拆分后

的结果是['12' , '2']，分别给 a 和 b 赋值后，a = '12' ， b = '2' ，字符串的大小比较是要一个字符一个字符来比较的。所以 a < b，本题错误。

第一轮 '12' < '2'
'1' < '2'

9. 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】本题主要考察元组的内容。元组 (tuple) 是不可变的数据类型，不支持 sort() 排序，调用 sort() 函数会引发 AttributeError 错误，要想对元组进行排序，可以使用 sorted() 函数，所以本题错误。

10. 以下 Python 代码执行后将输出 cherry。

```
lstA = ['grape', 'cherry', 'mango']
lstB = lstA
lstB.append('apple')
lstA.sort()
print(lstA[0])
```

【答案】错误

【解析】本题主要考察列表的特点。在定义好一个列表 lstA，然后让 lstB=lstA，这种新列表直接等于旧列表时，其实就是给 lstB 赋值，赋的值是 lstA 里储存的列表内容，这种赋值方法是引用赋值，即两者指向同一个内存，这样，改变一个列表另一个列表也会跟着改变。所以在在 lstB 添加元素 'apple' 后，lstA 中也会出现同一元素，此时再排序，根据字符串的排序特点，排序后的结果为 ['apple', 'cherry', 'grape', 'mango']，所以本题错误。

三、编程题

第 1 题

【问题描述】

老师带领同学们春游。已知班上有 N 位同学，每位同学有从 0 到 $N-1$ 的唯一编号。到了集合时间，老师确认是否所有同学都到达了集合地点，就让同学们报出自己的编号。到达的同学都会报出，但有的同学很顽皮，会多次报出。你能帮老师找出有哪些同学没有到达吗？

【输入描述】

输入包含 2 行。第一行包含两个整数 N 和 M ，表示班级有 N 位同学，同学们共有 M 次报出编号。约定 $2 \leq N, M \leq 1000$ 。

第二行包含 M 个整数，分别为 M 次报出的编号。约定所有编号都在合理范围内。

【输出描述】

输出一行。如果所有同学都到达，则输出 N ；否则由小到大输出所有未到达的同学编号，空格分隔。

【样例输入 1】

```
3 3
0 2 1
```

【样例输出 1】

```
3
```

【样例输入 2】

```
3 5
0 0 0 0 0
```

【样例输出 2】

```
1 2
```

【题目大意】

第一行输入两个数字，分别表示同学数量和报号次数，第二行表示同学们报的号，找出没有报出的号输出。

【解题思路】

本题主要考察输入多个数据如何处理，以及集合的内容。

1. 首先输入两个数字，通过拆分、转换、分别赋值。
2. 输入 M 个整数，拆分转换后，通过集合去重。
3. 创建集合，表示出正常的编号范围。
4. 比较，判断是否一致，根据要求完成输出。

【参考程序】

```
N, M = map(int, input().split()) #输入应到人数, 和报数次数
outSet = set(map(int, input().split())) #报数的集合, 已去重, M没有用上
allNumSet = set(range(N))

if outSet == allNumSet:
    print(N)
else:
    print(" ".join(map(str, sorted(allNumSet - outSet))))
```

第 2 题

【问题描述】

1. 网站注册需要有用户名和密码, 编写程序以检查用户输入密码的有效性。
2. 合法的密码只能由 a-z 之间 26 个小写字母、A-Z 之间 26 个大写字母、0-9 之间 10 个数字以及 !@#\$ 四个特殊字母构成。
3. 密码最短长度: 6 个字符, 密码最大长度: 12 个字符。
4. 大写字母、小写字母和数字必须有其中两种, 以及至少有四个特殊字符中的一个。

【输入描述】

1. 输入以英文逗号分隔的多个字符组合作为被检测密码。输入时的逗号都作为分隔符, 不被视作检验密码本身。
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. 逐行输出 0 个或多个合规的密码。
2. 输出顺序以输入字符串出现先后为序, 即先出现则先输出。

【样例输入 1】

```
seHJ12!@,sjdkffH$123,sdf!@^&12 HDH,123&^YUhg@!
```

【样例输出 1】

```
seHJ12!@
sjdkffH$123
```

【题目大意】

求有效密码问题，输入多个密码，每组密码之间用逗号隔开，判断密码是否合规，输出合规密码。

【解题思路】

本题主要考察字符串的遍历以及 ASCII 码的转换内容。

1. 首先输入多组密码，以逗号隔开，拆分出一个个独立的密码。
2. 遍历拆分后的列表，对遍历出的每一个字符串再次进行遍历，根据 ASCII 码的转换，分别求出小写字母、大写字母、数字、规定字符和其他字符的个数。
3. 根据求出来的个数确定密码里包含的字符种类，判断是否合规。
4. 按照要求完成输出。

【参考程序】

```
#密码合规检测
```

```
pwdList = input().split(",")
```

```
for itm in pwdList:
```

```
    lowerCount = 0
```

```
    upperCount = 0
```

```
    digitCount = 0
```

```
    specialCount = 0
```

```
    otherCount = 0
```

```
    for letter in itm:
```

```
        if ord('a') <= ord(letter) <= ord('z'):
```

```
            lowerCount += 1
```

```
        elif ord('A') <= ord(letter) <= ord('Z'):
```

```
            upperCount += 1
```

```
        elif ord('0') <= ord(letter) <= ord('9'):
```

```
            digitCount += 1
```

```
        elif letter in "!@#$%&':
```

```
            specialCount += 1
```

```
        else:
```

```
            otherCount += 1
```

```
            break
```

```
    triState = 0 #大写小写数字种类数量
```

```
    if lowerCount > 0:
```

```
    triState += 1
if upperCount > 0:
    triState += 1
if digitCount > 0:
    triState += 1

if triState >= 2 and specialCount >0 and (6 <= len(itm) <= 12) and
otherCount == 0:
    print(itm)
```