

GESP Python 四级样题卷

(满分：100分 考试时间：90分钟)

学校：\_\_\_\_\_

姓名：\_\_\_\_\_

题目	一	二	三	总分
得分				

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	A	B	A	C	C	D	A	C	D	B	C	D	A	A	C

1. 执行代码 `a, b = 5, 6; print(a or b)` 打印出的结果是 ( )

- A. 5
- B. 6
- C. True
- D. False

2. 水仙花数是指三位数中各位立方之和等于该数本身, 如  $153=1**3+5**3+7**3$ 。求出所有水仙花数的适合算法是 ( )

- A. 解析算法
- B. 枚举算法
- C. 排序算法
- D. 对分查找法

3. 执行下面 Python 代码, 其时间复杂度是 ( )。

```
tot = 0 #保存累加求和的结果
for i in range(1, 100+1):
    tot += i
```

```
print(tot)
```

- A.  $O(n)$
  - B.  $O(n \log n)$
  - C.  $O(n \log^2 n)$
  - D.  $O(n^2)$
4. Python 运算符 `in` 用于成员存在判断。假如成员数量相同且数量很大，效率最高的是( )。
- A. `list` 类型
  - B. `tuple` 类型
  - C. `set` 类型
  - D. `str` 类型
5. 在 Python 语言中，下列关键字不是专用于异常处理的是( )。
- A. `try`
  - B. `except`
  - C. `if`
  - D. `finally`
6. 在 Python 异常处理中，以下一定能得到全部执行的子句是 ( )
- A. `try` 子句
  - B. `except` 子句
  - C. `else` 子句
  - D. `finally` 子句
7. 下面关于 Python 自定义函数的说法中，错误的是 ( )
- A. Python 自定义函数中不可以含有两个或两个以上的 `return` 语句。
  - B. Python 自定义函数中可以没有 `return` 子句。
  - C. Python 自定义函数如含多个 `return` 语句，则每次调用最多执行一次 `return` 语句。

D. Python 自定义函数可以返回多个值，但本质上仍然是一个值。

8. 下列有关自定义函数的描述，正确的是（ ）。

A. 函数的定义必须在程序的开头。

B. 函数定义后，其中的程序可以自动执行。

C. 函数定义后需要调用才会执行，且可以多次调用。

D. 函数可以定义在调用之前，也可以定义在调用之后。

9. 下面 Python 代码执行后输出是（ ）。

```
lstA = [1, 2, 5, 6, 3, 7]
```

```
print(lstA.sort())
```

A. [1, 2, 3, 5, 6, 7]

B. [7, 6, 5, 3, 2, 1]

C. [1, 2, 5, 6, 3, 7]

D. None

10. 运行下列程序，输出的结果是？（ ）

```
def Snd(msg = 'XinXi'):
```

```
    print('XinXi' + msg)
```

```
Snd()
```

A. XinXi

B. XinXiXinXi

C. XinXimsg

D. 程序将提示运行错误

11. 运行如下 Python 代码，其输出是（ ）

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

```
b = (3, 4)
```

```
a = a + b
```

```
print(a)
```

- A. 1, 2
- B. (1, 2)
- C. (1, 2, 3, 4)
- D. 1, 2, 3, 4

12. 运行如下 Python 代码，其输出是（ ）。

```
a = [1, 2]
```

```
b = a
```

```
a.append(3)
```

```
print(a, b)
```

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

13. 下面 Python 函数用于判断整数 N 是否素数，效率最高的是（ ）。

```
import math
```

```
def isPrime(N):
```

```
    if N < 2:
```

```
        return False
```

```
    lastNum = _____ #此处填入合适代码
```

```
    for i in range(2, lastNum + 1):
```

```
        if N % i == 0:
```

```
            return False;
```

```
    return True
```

- A. `int(math.sqrt(N))`
- B. `N//2`
- C. `N`

D. 以上效率相同

14. 在 Python 中定义 Fx 的函数如下，调用错误的选择是（ ）。

```
def Fx(x, y=1, z=2):  
  
    return x+y+z
```

A. Fx()

B. Fx(1)

C. Fx(1, 2)

D. Fx(1, 2, 3)

15. 在下面 Python 代码中，填入正确的是（ ）。

```
from random import random as rnd  
  
print(_____) #填入正确代码
```

A. random()

B. random.random()

C. rnd()

D. random.rnd()

## 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	F	F	T	T	T	T	T	F	T	T

1. 在 Python 中, 只能顺序读取文件内容, 如读取 10 个字符, 必须先读取 1-9 个字符弃用, 然后读取第 10 个字符。( )
2. Python 可以读取文本文件, 但不能读取二进制文件。( )
3. 递推通常是通过计算前面的一些项来得出序列中的指定项的值。( )
4. 同样的问题使用不同的算法编程解决, 其时间复杂度可能很不一样。( )
5. 运行下面 Python 代码, 其输出为 6。( )

```
def Fx(a, b=2):  
    return a + b  
  
print(Fx(a=3, b=3))
```

6. Python 自定义函数可以返回多个值, 但本质上是一个值。( )
7. 对于斐波那契数列: 1, 1, 2, 3, 5, 8……, 常采用递推方式求解, 但也可用非递推方式。( )
8. 在 Python 自定义函数中, 如果局部变量与全局变量重名, 则修改局部变量的值将同步修改全局变量的值。( )
9. Python 中自定义函数时, 可不声明形参的数据类型, 由调用自定义函数时传递的实参类型确定。( )
10. 下列程序段返回的值为"Hello!Python" ( )。

```
Hello = "Hello!Python"  
def Foo():  
    Hello = "Hello!"  
    return Hello  
Foo()  
print(Hello)
```

三、编程题 (每题 25 分, 共 50 分)

题号	1	2
答案		

1. 【问题描述】

角谷猜想是正整数  $N$  如果为偶数则将其变换为二分之一, 如果为奇数则将其变换 3 倍加 1, 重复该过程将最终变为 4、2、1 的重复。到目前为止, 该猜想尚未在数学上得到证明。编写程序输出指定范围内变化步骤最多的数及其变化步骤总数和变化过程。对于 13 其变化步骤为: 13, 40, 20, 10, 5, 16, 8, 4, 2, 1。

根据上述描述, 编写程序。在样例输出中, 18 表示介于 10 和 20 之间的 18 变化步骤最多, 共计变换 21 步, 其变换过程如最后行所示。变化步骤部分的箭头由英文字符减号和大于号联合构成, 每个数之后均有。

【样例输入】

10, 20

【样例输出】

18

21

18->9->28->14->7->22->11->34->17->52->26->13->40->20->10->5->16->8->4->2->1->

2. 【问题描述】

现有  $n$  个人围成一圈玩游戏, 从第 1 个人开始报数, 数到第  $m$  个人出局, 然后从出局的下一个人开始报数, 数到第  $m$  个人又出局, ..., 如此反复到只剩下最后一个是胜利者。设  $n$  个人的编号分别为 1, 2, ...,  $n$ , 打印出局的顺序。

根据上述描述, 编写程序

【样例输入】

10

4

【样例输出】

4  
8  
2  
7  
3  
10  
9  
1  
6

WINNER5

参考代码:

```
def Fx(n, k):  
    L = list(range(1, n + 1))  
    if n == 1:  
        print(f'WINNER{L[0]}')  
    else:  
        x = 0  
        while len(L) != 1:  
            x = (x + k) % len(L) - 1  
            print(L[x])  
            del L[x]  
            if x < 0:  
                x = 0  
            print(f'WINNER{L[0]}')  
m = int(input())
```

```
n = int(input())
```

```
Fx(m, n)
```