

PAAT_Python编程中级2练习题一

一、单项选择题（共 20 题，每题 3.5 分，共 70 分）

P1_1. 下列关于 Python 代码复用和程序抽象的叙述中，正确的是（ ）。

- A. 代码复用可以提高程序的运行效率
- B. 对程序功能进行分解和抽象，不利于大型应用程序的实现
- C. 函数是代码复用的一个重要组成部分
- D. 代码复用和程序抽象增加了编程难度，应尽量避免使用

P2_1. 下列关于 Python 函数的叙述中，正确的是（ ）。

- A. 用户自定义函数可以使用 `define` 定义函数
- B. 用户自定义函数可以没有形参，但不能没有函数名
- C. 用户自定义函数名不能和 Python 内置函数名重名
- D. 用户自定义函数中最少有一条返回值的 `return` 语句

P3_1. 在 Python 中，运行下方函数定义代码段后，下列调用结果正确的是（ ）。

```
def demo(a, b):  
    c = a + b  
    return c
```

- A. `demo(2, 3)` 的值为 5
- B. `demo('2', '3')` 的值为 '5'
- C. `demo(2, 3)` 的值为 '5'
- D. `demo('2', '3')` 的值为 5

P4_1. 在 Python 中，运行下方函数定义代码段后，下列调用结果正确的是（ ）。

```
def demo(str_t, n=2):  
    str_tmp = ""  
    for c in str_t[::n]:  
        str_tmp = str_tmp + c  
    return str_tmp
```

- A. `demo("0123456789")` 的值为 '0123456789'
- B. `demo("0123456789", 2)` 的值为 '13579'
- C. `demo()` 的值为 None
- D. `demo("0123456789", 3)` 的值为 '0369'

P5_1. 下列关于递归函数的叙述中，正确的是（ ）。

- A. 递归函数应该有边界条件以保证函数正确性
- B. 递归函数必须用函数名作为返回值
- C. 递归函数的实现通常比非递归函数复杂
- D. 递归函数中必须包含循环结构

P6_1. 在 Python 中, 运行下方函数定义代码段后, 下列调用结果正确的是 ()。

```
def demo(n):  
    if n <= 0:  
        return 1  
    elif n in (3, 5):  
        return 8  
    else:  
        return n + demo(n-3)
```

- A. demo(8) 的值为 18
- B. demo(6) 的值为 16
- C. demo(5) 的值为 8
- D. demo(4) 的值为 5

P7_1. 下列关于 Python 文件操作的叙述中, 不正确的是 ()。

- A. open 是 Python 中打开文件的内置函数
- B. 文件使用完毕, 应使用打开文件对象的 close() 方法将其关闭
- C. Python 能够以文本形式或二进制形式打开文件进行操作
- D. Python 打开的文件不能既进行读操作, 又进行写操作

P8_1. 文件 a.txt 中的一行数据为:

Hello,world.

运行语句 f=open("a.txt","rt") 后, 下列叙述正确的是 ()。

- A. f.readall() 的值为 'Hello,world.'
- B. f.readlines() 的值为 'Hello,world.'
- C. f.readable() 的值为 True
- D. f.readline(1) 的值为 'Hello,world.'

P9_1. 下列代码段可以将古诗“锄禾日当午, 汗滴禾下土。谁知盘中餐, 粒粒皆辛苦。” 5 字一行带标点写入文件 a.txt 的是 ()。

A.

```
poem = ["锄禾日当午, ", "汗滴禾下土。", "谁之盘中餐, ", "粒粒皆辛苦。"]  
f = open("a.txt", "wt")  
for line in poem:  
    f.write(line)  
f.close()
```

B.

```
poem = ["锄禾日当午, ", "汗滴禾下土。", "谁之盘中餐, ", "粒粒皆辛苦。"]  
f = open("a.txt", "w")  
for line in poem:  
    f.write(line+"\n")  
f.close()
```

C.

```
poem = ["锄禾日当午，", "汗滴禾下土。", "谁之盘中餐，", "粒粒皆辛苦。"]  
f = open("a.txt", "rt")  
for line in poem:  
    f.write(line+"\n")  
f.close()
```

D.

```
poem = ["锄禾日当午，\n", "汗滴禾下土。 \n", "谁之盘中餐， \n", "粒粒皆辛苦。 \n"]  
f = open("a.txt", "wt")  
f.write(poem)  
f.close()
```

P10_1. 同一目录下仅有两个文件 a.py 和 b.py, a.py 中只有一行代码 a = 20, 下列代码段可以作为 b.py 内容, 运行后输出值为 100 的是 ()。

A.

```
from a.py import *  
print(a * 5)
```

B.

```
from a import *  
print(a * 5)
```

C.

```
from a import *  
b = 5  
print(a * b.b)
```

D.

```
import a.py as a  
print(a * 5)
```

P11_1. 下列关于 Python 面向对象编程的叙述中, 正确的是 ()。

- A. Python 使用内置函数 class 定义类
- B. Python 中的所有内容都可以称为对象
- C. 对象是属性、函数和方法的封装
- D. 对象与对象之间的交互称为命令调用

P12_1. 运行下方代码段，下列关于使用对象的属性或方法结果正确的是（ ）。

```
class Car(object):
    def beep(self):
        print("Beep! beep!")
    def __init__(self, color):
        self.color = color
    def prt_color(self):
        print(self.color)
car1 = Car("red")
```

- A. 执行 `print(car1.color)` 得到的输出结果为 red
- B. 执行 `print(car1.beep)` 得到的输出结果为 Beep! Beep!
- C. 执行 `car2 = Car()`，可以得到 Car 类的新实例 car2
- D. 执行 `car1.color("blue")` 可以修改 car1 的 color 属性

P13_1. 下列关于 Python 中模块和包的叙述中，不正确的是（ ）。

- A. 类和函数均可放入模块组织成包供 Python 编程使用
- B. Python 通过构建包可以组织多文件结构，统一功能应用方式
- C. Python 包是一个含有 `__init__.py` 文件的文件夹
- D. Python 包文件夹中不允许有子文件夹

P14_1. 运行下方代码段，输出的结果是（ ）。

```
a = 100
def demo(v):
    global a
    a = 200
    print(a + v)
print(a)
demo(a)
print(a)
```

- A.
100
200
200
- B.
100
300
200
- C.
200
300
200

D.
100
200
100

P15_1. 下列关于 pip 工具的叙述中, 不正确的是 ()。

- A. pip list 命令可以列出当前 Python 环境已安装的第三方库名称和版本
- B. 使用 pip install 安装第三方库时, -i 参数可以指定库的来源网址
- C. 使用 pip 工具可以安装 .whl 文件类型的第三方库
- D. pip 工具不能卸载库, 需要用户将第三方库从 Python 文件夹下删除

P16_1. 下列关于 Python 第三方库 jieba 的叙述中, 不正确的是 ()。

- A. jieba 库是当前应用广泛的一个中文分词第三方库
- B. jieba.cut(s) 方法可以将字符串 s 分词后, 返回列表型数据
- C. jieba.add_word(w) 方法可以将词 w 加入 jieba 本地分词词典
- D. jieba.del_word(w) 方法可以将词 w 从 jieba 本地分词词典中移除

P17_1. 运行下方代码段, 下列叙述正确的是 ()。

```
a = eval("5 + 5")  
b = 5.0  
c = eval("b") * 5  
d = abs(b * -b)
```

- A. 变量 a 的数据类型为字符串型
- B. 变量 d 的值为 25
- C. 变量 c 的值为 'bbbbbb'
- D. 变量 b 和 d 的数据都是浮点型

P18_1. 运行下方代码段, 输出的结果是 ()。

```
t01 = tuple("12311234556")  
s01 = set(t01)  
print(len(t01))  
print(len(s01))  
print(4 in s01)
```

A.
11
6
False

B.
11
6
True

C.
1
11
True
D.
1
6
True

P19_1. 运行 `import random` 语句后, 下列叙述不正确的是 ()。

- A. `random.randint(60, 100)` 返回一个 60 (含) 到 100 (含) 间的随机整数
- B. `random.randrange(60, 101)` 返回一个 60 (含) 到 100 (含) 间的随机整数
- C. `int(random.random()*40)+60` 返回一个 60 (含) 到 100 (含) 间的随机整数
- D. `random.choice(range(60,101))` 返回一个 60 (含) 到 100 (含) 间的随机整数

P20_1. 运行 `import time` 语句后, 下列叙述正确的是 ()。

- A. `time.time()` 返回值为自 1970 年 1 月 1 日 00:00:00 到此刻的秒数
- B. `time.ctime()` 以中文形式返回当前的日期时间
- C. `time.sleep(n)` 可以让程序暂停 n 毫秒
- D. `time` 库是 Python 中较常用的第三方库

二、编程题 (共 3 题, 每题 10 分, 共 30 分)

P21_1. 编写程序实现求解水仙花数问题, 要求如下:

1. 使用两次 `int(input())` 接收用户依次输入的两个 3 位数字整型数据;
2. 求解第一个数字 (含) 到第二个数字 (含) 之间的所有的水仙花数;
3. 按由小到大顺序将找到的每个水仙花数输出, 输出时每个水仙花数占一行。

说明:

(1) 水仙花数是指一个 3 位数, 它的每个位上的数字的 3 次方之和等于它本身, 例如:
 $1^{**3} + 5^{**3} + 3^{**3} = 153$, 因此 153 是水仙花数;

注: 一个数字 a 的 3 次方, 指的是 3 个 a 的积的运算, Python 中可表达为 `a*a*a`
或 `a**3`。

- (2) `input()` 函数中不要增加任何提示用参数;
- (3) 输出结果不要使用任何空格等空白字符修饰。

样例:

输入:

371

500

输出:

371

407

P22_1. 编写程序，使用自定义函数实现求第 n 项斐波那契数列值，要求如下：

1. 函数名称为：fib ；
2. fib 函数的参数：一个，自然数，参数名称不限；
3. fib 函数的返回值：一个，非负整数，函数参数对应项的斐波那契数列值。

考生书写和提交代码注意事项：

1. 考生请勿修改或删除代码编辑窗口中已存在的内容，如下方代码段：

```
# 你的代码必须写在在本行上方，上方可增加任意行，下方代码必须保留 #  
eval(input())  
# 这是你提交代码的最后一行，本行后不允许书写任何代码 #
```

2. 考生书写的代码必须在代码窗口原有 3 行代码上方；
3. 考生提交的代码只应为函数的定义和功能，不应包含任何输入、输出语句及测试代码。

说明：斐波那契数列指的是这样一个数列 1, 1, 2, 3, 5, 8, 13, 21, 34 …… 这个数列的第 1 项和第 2 项均为 1，从第 3 项开始，每一项都等于前两项数值之和。例如，第 6 项斐波那契数列值为第 4 项和第 5 项的值 3 和 5 相加之和，即 8。

函数调用举例说明：

fib(6) 的返回值应为 8。

P23_1. 修改并调试已提供的代码，使其正确运行并完成如下功能：

1. 引入 random 库；
2. 使用 int(input()) 接收用户输入的一个整型数据；
3. 将用户输入的整型数据作为 random 库的 seed() 方法的参数，设定随机数种子；
4. 使用 random 库的 randint() 方法依次随机生成 10 个介于 10（含）到 100（含）的整数；
5. 输出这 10 个随机数中的最小值。

说明：

(1) 整个程序运行期间，random 库的 randint() 方法执行次数应为 10 次，不应多于或少于该次数。

(2) 考生请勿修改并调整编辑器中原有代码内容和顺序

样例：

输入：

30

输出：

61

原始代码如下：

```
# 考生在下一行引入 random 库
```

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

```
# 考生在下一行使用 random 库的 seed() 方法，按题目要求初始化种子
```

```
# 考生在下方书写代码，完成依次生成 10 个随机整数，并输出最小值操作
```

```
# 生成的随机整数范围应介于 10（含）到 100（含）之间
```