

# Python 程序设计基础互助课堂

## 期末复习 1

易翔

注：本章练习中的所有语法规则、代码执行结果等对 python 版本存在依赖关系的内容，均以 python==3.10.9 为准。

2023 年 6 月 5 日

### 一. 代码 本大题共 6 小题

1. 下列代码中存在 1 处错误，请指出程序运行时抛出异常的行号，出现异常的原因，并写出修改后的代码。

```
1 # 示例：学生考试成绩统计
2 students = {"Alice": 85, "Bob": 90, "Charlie": 75, "David": 92, }
3 # 任务背景：计算学生的平均分和最高分
4 total_score = 0
5 max = max(students.values())
6 for score in students.items():
7     total_score += score
8
9 count = len(students)
10 # 避免除以 0 的错误
11 average_score = total_score / count if count > 0 else 0
12
13 # 输出结果
14 print("学生人数:", count)
15 print("最高分:", max)
16 print("平均分:", average_score)
```

2. 下列是一个用于判断某个日期是否为合法日期的代码，其中存在若干的**语法错误**和**逻辑错误**，请先指出存在语法错误的行号，修改语法错误；随后指出存在逻辑错误的行号，修改逻辑错误。

```
1 def is_valid_date(date_string)
2     year, month, day = map(int, date_string.split('-'))
3     # 判断年份是否合法
4     if year < 1000 or year > 9999
5         return False
6     # 判断月份是否合法
7     if month < 1 or month > 12
8         return False
9     # 判断天数是否合法
10    if month == 1 or 3 or 5 or 7 or 8 or 10 or 12 # 大月，有31天
11        if day < 1 or day > 31
12            return False
13    elif month == 4 or 6 or 9 or 11 # 小月，有30天
14        if day < 1 or day > 30
15            return False
16    elif month == 2 # 二月
17        if is_leap_year(year) # 闰年
18            if day < 1 or day > 29
19                return False
20        else # 非闰年
21            if day < 1 or day > 28
22                return False
23    return True
24 def is_leap_year(yyyy)
25     return (yyyy % 4 == 0 and yyyy % 100 != 0) or yyyy % 400 == 0
26 print(is_valid_date(input()))
```

3. 在尝试计算斐波那契数列时，我们有时会使用下列代码来完成目标。但是在所求的项数过大时，往往无法在有限的时间内得到我们所需要的结果。请尝试修改下面的代码，将递归调用的方式改为利用循环求解的方式。

```
1 def calculate_fibonacci (n):  
2     if n < 2:  
3         return 1  
4     else:  
5         return calculate_fibonacci (n-1) + calculate_fibonacci(n-2)  
6  
7 print( calculate_fibonacci (1000))
```

4. 请写出下列代码的输出。

```
1 def concat(*args, sep="/"):
2     return sep.join ( args )
3
4 print(concat("foobar"))
5 print(concat("earth", "mars", "venus"), end="\n")
6 print(concat("earth", "mars", "venus", sep="."))
7 print(concat("a", "b", { 'sep': '!'}))
```

5. 下面的代码尝试找出两个单词中同时出现的辅音字母。请指出代码中存在**语法错误**的行号，并更正该错误。

```
1 a = {} # 创建一个空的集合
2 b = {x for x in 'language' if x not in 'aeiou'}
3 c = {x for x in 'python' if x not in 'aeiou'}
4 for element in b & c:
5     a.add(element)
6 print(a)
```

6. 扫雷是一个十分有意思的游戏。在传统的扫雷中，格子中的数字仅代表了**相邻**地雷的数量，现将格子中数字含义改为距离当前方格周围  $n * n$  区域内地雷的数量。请修改已有的代码以实现该功能。

```
1 import numpy as npy
2 def check(array, n):
3     """
4         array: Game environment, also known as board. A numpy array.
5         n: int
6     """
7     result = npy.zeros(array.shape)
8     # complete your code here
9
10    # end
11    return result
```

e.g. array

|    |    |    |    |   |
|----|----|----|----|---|
| -1 | 0  | -1 | -1 | 0 |
| 0  | 0  | 0  | -1 | 0 |
| -1 | 0  | 0  | 0  | 0 |
| 0  | 0  | 0  | 0  | 0 |
| 0  | -1 | 0  | 0  | 0 |

在  $n=2$  时，result 矩阵应为

|    |    |    |    |   |
|----|----|----|----|---|
| -1 | 4  | -1 | -1 | 3 |
| 3  | 5  | 5  | -1 | 3 |
| -1 | 6  | 6  | 4  | 3 |
| 2  | 3  | 3  | 2  | 1 |
| 2  | -1 | 2  | 1  | 0 |