**考试说明**

1. **考试时间：合计180分钟，选择题部分30分钟交卷，允许提前交卷。**
2. **考试过程中，不能连接未经指定网站或服务器。**
3. **闭卷考试部分，不能查阅任何类型的参考资料。**
4. **开卷考试部分，可以查阅纸质文档，不能查阅除Python编程环境自带帮助文件以外的任何类型的电子文档。**
5. **考试过程中，不得使用任何形式的电子存储设备，不可使用手机。**
6. **违反上述2-5条者，视为考试作弊。**

**选择题答题方式（25分，闭卷，严禁使用python编程环境进行尝试）**

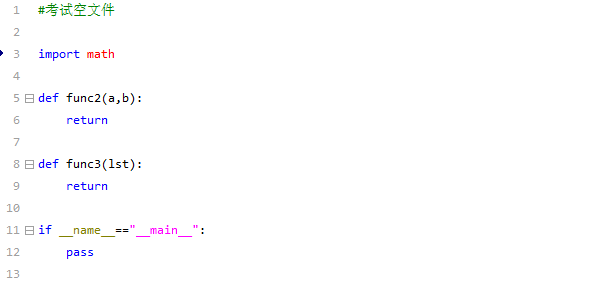
1. **打开浏览器，在地址栏中输入**http://192.168.125.3**，点击相应链接进入登录页面。**
2. **按要求输入两遍自己的学号。**
3. **点击“登录”按钮即可进入答题页面。如考试尚未开始，系统会进入等待页面并倒计时。考试开始时间到，系统会自动进入答题页面。**
4. **在页面左侧选择题号，页面右侧即会显示相应的题目。考生只需点击选择相应的选项。**
5. **答题过程中如关闭浏览器或出现系统故障导致计算机重新启动，系统不会丢失之前已经完成的题目的答案。考生可以打开浏览器重新登录并继续考试。**
6. **答题完成后，点击“交卷”按钮即可完成交卷。交卷后不能再次登录系统继续考试。**
7. **考试结束时间到，系统会自动收卷。**

**编程题注意事项与提交方式（75分，开卷）**

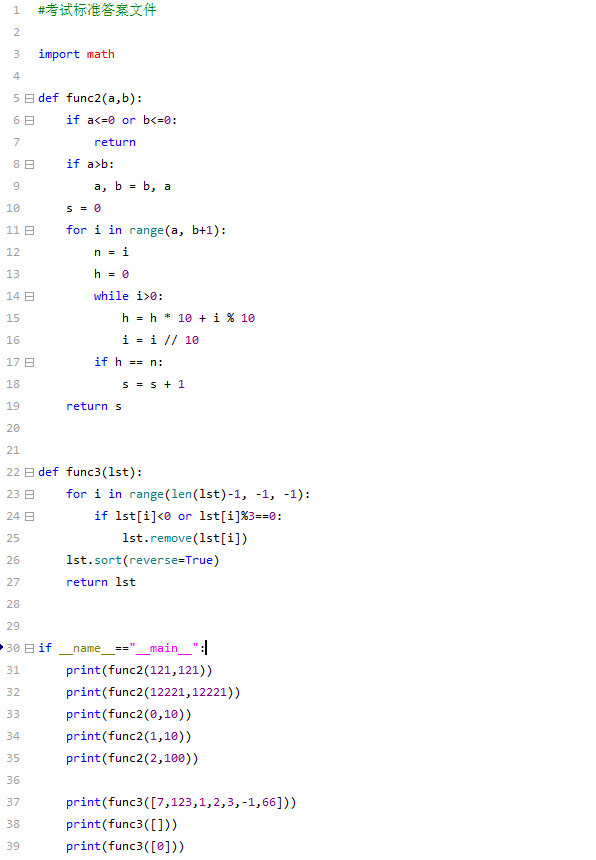
1. **下载sample.py，保存到D盘。严禁将代码文件或数据文件保存在C盘中或桌面上。**
2. **根据题目要求补全sample.py中的空函数。sample.py中的已有函数的函数名、参数数量和顺序不可以修改。**
3. **调用自己写的函数、自己测试的代码等，请写入到if \_\_name\_\_=="\_\_main\_\_":，不要写到全局环境中。**
4. **不要在全局环境中调用input()，否则会导致阅卷失败**
5. **每个函数中不需要使用print输出结果，而是用return返回结果。**
6. **不要使用关键字（if、else、 break、 def等，会导致语法错误）作为自己的变量名，也不要使用内嵌名字（如list、 int、 input等，没有语法错误，但是很危险）作为自己的变量名，例如：list = []**
7. **代码中不要出现任何中文。**
8. **确保提交的时候，自己的程序可以正常运行，不要遗留任何语法错误。**
9. **所有脚本程序内容必须仅包含在一个脚本程序文件（py文件）中。**
10. **提交前务必关闭IDLE或PyCharm编程环境。**
11. **在浏览器的地址栏中输入**http://192.168.125.3**，点击相应链接进入提交页面。**
12. **按要求输入两遍自己的学号。**
13. **点击“选择文件”按钮，选择自己的脚本程序文件。点击“提交”按钮提交。**
14. **如提交成功，系统会显示相关信息。如果提交不成功，请重复步骤24-26。**
15. **提交成功后，可点击“查看内容”按钮检查提交的内容**

**答题方式示意图：**

1. **请根据图1、图2和图3所示的说明严格规范源文件结构。**



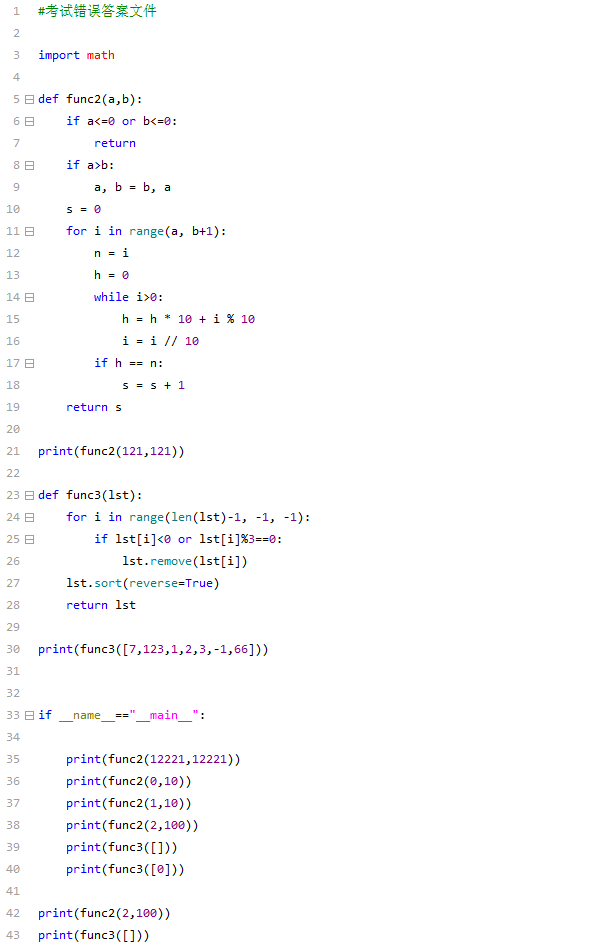
**图1 空文件样例**



每道题目的代码包含在一个确定名称的函数体内。函数中不包含input函数和print函数。

测试代码写在此处，且全部包含的if语句体内，即保持相同的缩进位置。

**图2 正确的答题文件格式**



错误！！！不要在if语言体外插入测试代码

错误！！！不要在函数之间插入测试代码

错误！！！不要在函数之间插入测试代码

**图3 错误的答题文件格式**

**题目说明**

1. 判断点的区域：在XY坐标平面内有一个正方形，其4个角点的坐标分别是（1，-1），（1，1），（-1，-1），（-1，1）。写一个程序，判断一个给定的点和这个正方形的关系。上述所有坐标值都为整型。如在正方形内则返回1，在边界上返回0，在正方形外返回-1。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 由两个整数分别表示点的X坐标和Y坐标 |
| 输出要求 | 根据判断结果，按照题目中所述要求返回相应数值 |
| 其它要求 | 将代码写入函数func1 |

建议测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| 0,0 | 1 |
| 1,0 | 0 |
| 2,2 | -1 |

1. 统计数字出现次数：请统计在某个给定闭区间范围[L, R]的所有整数中，数字d出现的次数。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 3个**肯定非负**的整数。第1个表示数字d，第2个表示判断区间的左边界L，第3个表示判断区间的右边界R。 |
| 输出要求 | 以整数的形式返回出现次数。  如d不在0~9的范围内，返回None；  如**L大于等于R导致不能构成有效区间**时，则返回None。 |
| 其它要求 | 将代码写入函数func2 |

建议测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| 0,0,11 | 2 |
| 1,3,4 | 0 |
| 1,4,4 | None |
| 12,3,6 | None |

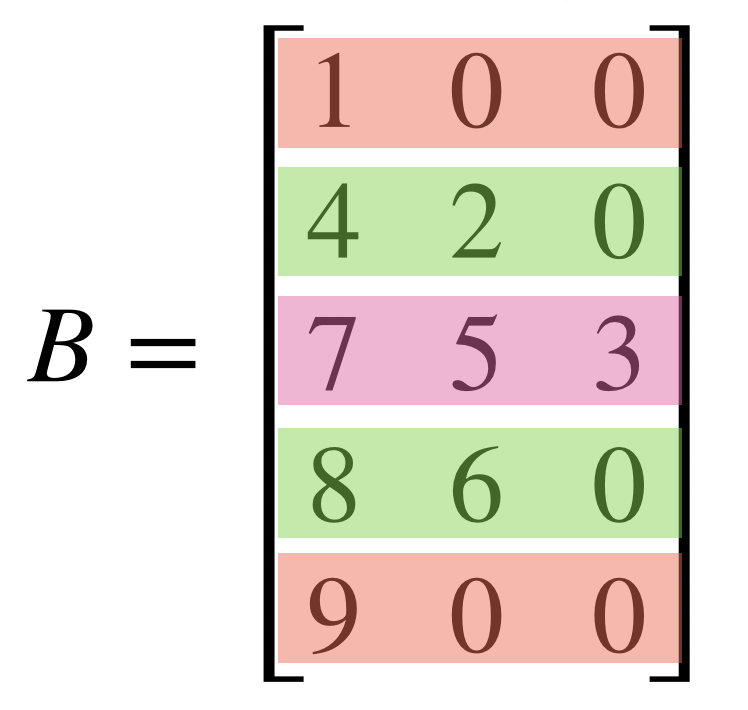
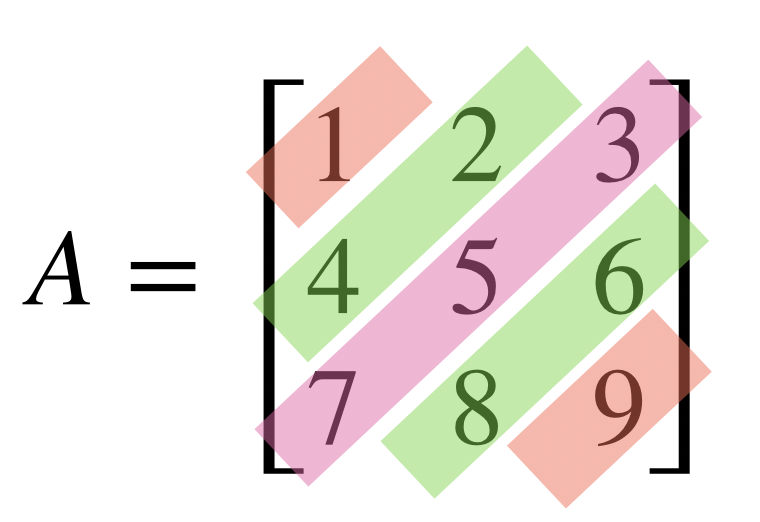
1. 一个正整数的头部和尾部分别是其第一位数字和最后一位数字。比如123的头部是1，尾部是3。5的头部和尾部都是5。给定一个正整数列表，将其中每个元素用它的头部和尾部进行替换，从而得到一个包含若干数字的列表T，将T中的质数保持不变并看成分隔符，可以把T分割成若干个子序列，对于每个子序列，将其中的数字进行合并得到一个新的数，返回合并之后的列表。比如，对于列表[1, 234, 5, 6, 70, 890]，替换之后的列表是[1, 1, 2, 4, 5, 5, 6, 6, 7, 0, 8, 0]，其中包含的子序列有<1,1>, <4>, <6,6>和<0,8,0>，合并之后的列表是[11, 2, 4, 5, 5, 66, 7, 80]。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 仅包含正整数的列表 |
| 输出要求 | 列表 |
| 其它要求 | 将代码写入函数func3 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [1, 234, 5, 6, 7, 890] | [11, 2, 4, 5, 5, 66, 7, 7, 80] |
| [12, 34, 56, 78, 90] | [1, 2, 3, 4, 5, 6, 7, 890] |
| [123] | [1, 3] |

1. 将一个m行m列的矩阵A按照对角线转置成一个2m-1行m列的矩阵B，即A的第i条对角线上的元素成为B的第i行上的元素（为保证B的每行都有m列，空缺位置用0填充）。下面是一个3行3列矩阵A按照对角线转置的结果。



在上面的例子中，A的第0条对角线有1个元素（1），第1条对角线有2个元素（4，2），第2条对角线有3个元素（7，5，3），第3条对角线有2个元素（8，6），第4条对角线有1个元素（9）。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 以嵌套列表表示的矩阵 |
| 输出要求 | 以嵌套列表表示的转置后的矩阵 |
| 其它要求 | 将代码写入函数func4 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [[1,2,3], [4,5,6], [7,8,9]] | [[1,0,0], [4,2,0], [7,5,3], [8,6,0], [9,0,0]] |
| s[[1,2], [3,4]] | [[1,0], [3,2], [4,0]] |

1. 给定一个英文单词w。w由大小写字母构成，不含别的字符。如果w中存在大写字母，则将w改成首字母大写，其它字母都小写的形式。如果w中不存在大写字母，将w改成首、尾字母大写，其它字母都小写的形式。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | w是一个英文单词，符合题中所述形态。 |
| 输出要求 | 如单词长度小于2个字符，返回None。否则，按题目要求返回。 |
| 其它要求 | 将代码写入函数func5。 |

建议测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| “wE” | “We” |
| “comPUtER” | “Computer” |
| “university” | “UniversitY” |

1. 给定一个字符串s，由单词和分隔符构成。其中，单词由大小写字母和数字符构成。除英文字母和数字符以外的字符均视为分隔符。请将给定的字符串s转换成每个单词之间**仅有一个空格**分隔的形式，最后一个单词后不添加空格。如单词长度大于5个字符，则将单词除首尾字符以外的字符均改成“\*”。单词的次序不变。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | s为给定的字符串，符合题中所述形态。 |
| 输出要求 | 如s中不包含单词，则返回空字符串。否则，按题目要求返回。 |
| 其它要求 | 将代码写入函数func6。 |

建议测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| "hello ! world.,Computer. class,54,5w" | "hello world C\*\*\*\*\*\*r class 54 5w" |
| "one two:::three, FOUR.., five" | "one two three FOUR five" |
| "worldcomputer.," | "w\*\*\*\*\*\*\*\*\*\*\*r" |
| "!=-,." | "" |

1. 现有一个列表words，其中存储了n个字符串，每一个字符串是一个单词，另有一个字符串 chars，其中存储了m个字母（字母区分大小写）。假设小明用 chars 中的字母拼写出 words 中的某个单词，那么就认为小明掌握了这个单词。注意：每次拼写时，chars 中的每个字母都只能用一次。返回words 中小明所掌握的单词个数。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | words中的每一个字符串长度一定大于等于1  chars的长度一定大于等于1 |
| 输出要求 | 返回一个整数 |
| 其它要求 | 将代码写入函数func7 |

建议测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| words = ["cat","bt","hat","tree"], chars = "atach" | 2  解释：  可以形成字符串 "cat" 和 "hat" |
| words = ["hello","world","soochow"], chars = "welldonehoneyr" | 2  解释：  可以形成字符串 "hello" 和 "world" |

1. 现有一个列表lst，其中有若干个元素，每个元素是一个元组，每个元组包含一个字符串和一个整数，字符串存了学号（**合法学号是9位数字符号**），整数存储了志愿者活动的时间(单位是小时，每次志愿者活动的**合法时间是1-3小时之间**）。请编写程序，去除**学号或者时间不合法的记录**，然后计算出每位同学的志愿者活动总时间， 按照总时间从大到小排序，时间相同的时候按照学号升序排序。最后以元组的形式返回排名第一的同学学号和志愿者活动的总时间。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | lst是一个元素是元组的列表，每个元组的时间元素一定是整数 |
| 输出要求 | 如果没有满足条件的结果返回None，否则返回一个元组，元组的第一个元素是学号，第二个元素是总时间 |
| 其它要求 | 将代码写入函数func8 |

建议测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [('192740506',3),  ('192740101',2)  ('192740101',2)] | ('192740101', 4)  解释：  '192740101'的总时间最大。 |
| [('192740506',3),  ('192740A01',2),  ('192740101',3)] | ('192740101', 3)  解释：  第二个元组中学号不合法，'192740101'的时间和'192740506'时间相同，但是学号更小 |
| [('19274056',3),  ('192740A01',2),  ('192740101',3)] | ('192740101', 3)  解释：  有两个不合法学号'19274056'和'192740A01' |