**考试说明**

1. **考试时间：合计150分钟，选择题部分不超过60分钟，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（20分，闭卷，自动阅卷，严禁使用python编程环境进行尝试）**

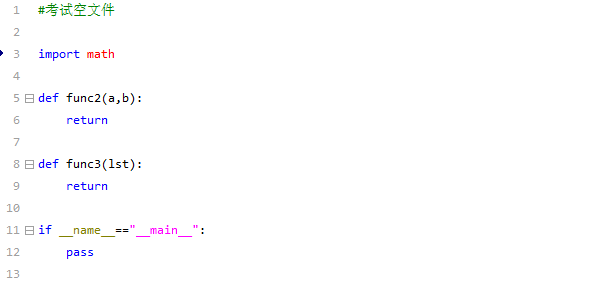
1. **打开浏览器，在地址栏中输入**http://192.168.125.3**，点击相应链接进入登录页面。**
2. **按要求输入两遍自己的学号。**
3. **点击“登录”按钮即可进入答题页面。如考试尚未开始，系统会进入等待页面并倒计时。考试开始时间到，系统会自动进入答题页面。**
4. **在页面左侧选择题号，页面右侧即会显示相应的题目。考生只需点击选择相应的选项。**
5. **答题过程中如关闭浏览器或出现系统故障导致计算机重新启动，系统不会丢失之前已经完成的题目的答案。考生可以打开浏览器重新登录并继续考试。**
6. **答题完成后，点击“交卷”按钮即可完成交卷。交卷后不能再次登录系统继续考试。**
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（80分，开卷，自动阅卷）**

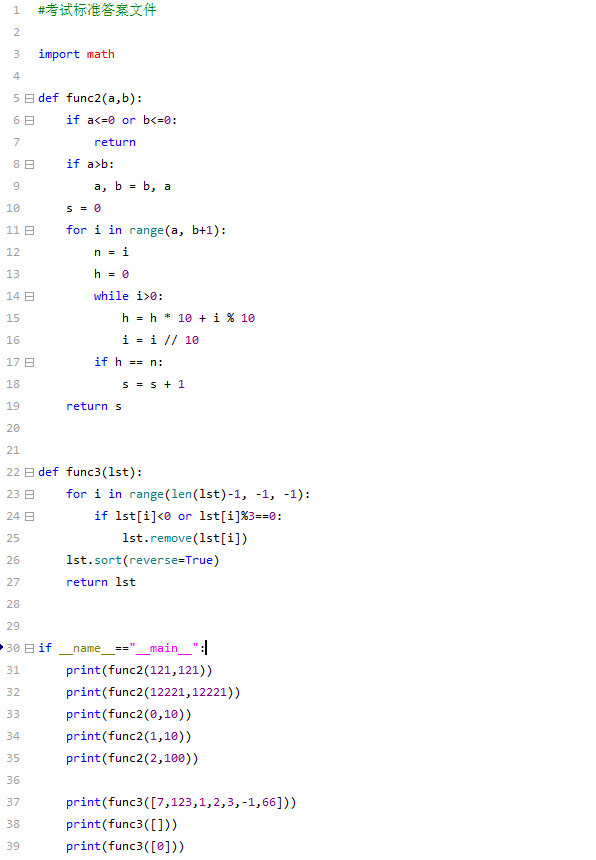
1. **下载sample.py，保存到D盘。严禁将代码文件或数据文件保存在C盘中或桌面上。**
2. **根据题目要求补全sample.py中的空函数。sample.py中的已有函数的函数名、参数数量和顺序不可以修改。**
3. **测试代码，请写入到if \_\_name\_\_=="\_\_main\_\_":部分，不要写到全局环境中。**
4. **不要在代码中调用input()，否则会导致阅卷失败。如阅卷失败，编程题部分不得分。**
5. **每个函数中不需要使用print()输出结果，而是用return返回结果。如用print()输出结果将导致该函数不得分。**
6. **不要使用关键字（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. **提交的时候，确保程序可以正常运行，不要遗留任何语法错误。语法错误、程序异常等将导致阅卷失败。如阅卷失败，编程题部分不得分。**
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. **确保文件已经存盘，并知道文件存放的位置。提交前务必关闭IDLE或PyCharm编程环境。如文件丢失或无法找到，编程题部分不得分，不能重考。**
11. **考试过程中，请经常进行存盘操作，以防突发异常情况而导致程序未保存。**
12. **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。**
13. **按要求输入两遍自己的学号。**
14. **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。**
15. **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25-27。**
16. **提交成功后，可点击“查看内容”按钮检查提交的内容。**
17. **编程题部分由阅卷系统自动批阅，以运行正确的测试用例数计分。不采取人工阅卷的方式。**

**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。**



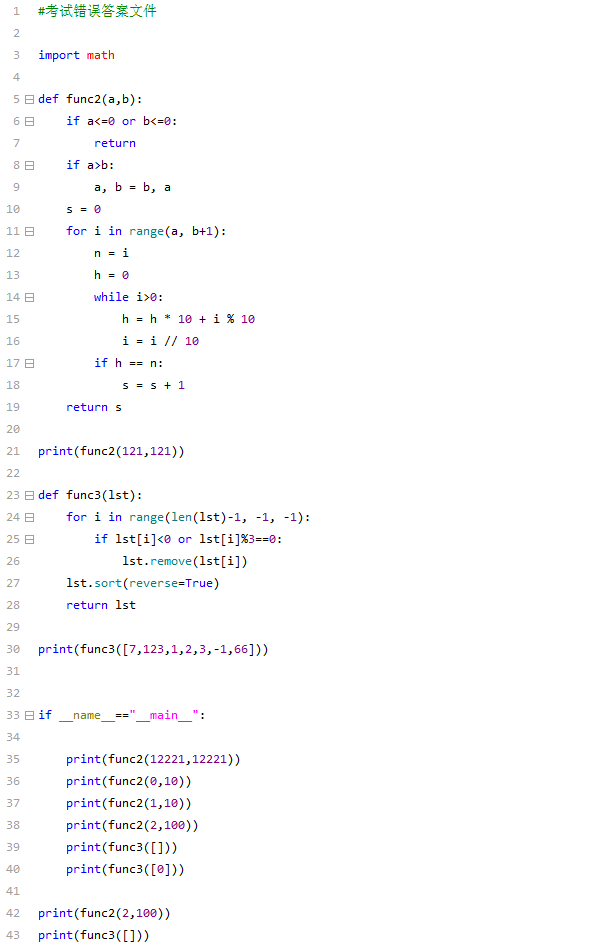
**图1 空文件样例**



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。

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。

**图2 正确的答题文件格式**



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

错误！！！不要在if语句体外插入测试代码

**图3 错误的答题文件格式**

**题目说明**

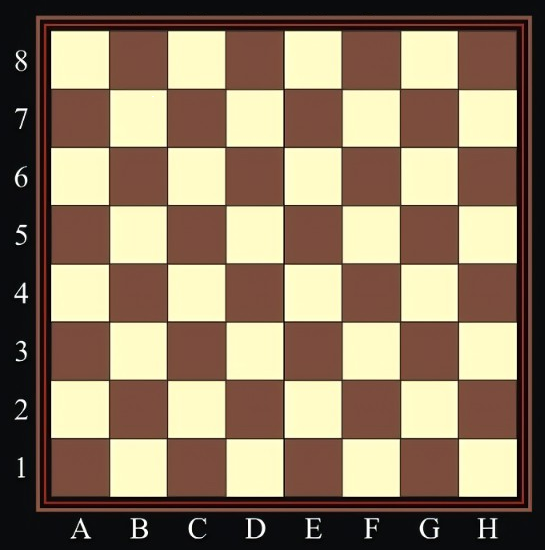
1. 现有一个列表lst1，其中存储了冬酿酒和藏书羊肉的价格，单位是元（一定是整数），第二个列表lst2依次存储了购买冬酿酒和藏书羊肉的数量（一定是整数），请计算出总价。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | lst1和lst2分别有2个整型元素，每个元素的值都>=0 |
| 输出要求 | 返回总价 |
| 其它要求 | 将代码写入函数func1。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [5,105] , [2,1] | 115 |
| [8,110] , [0,1] | 110 |

1. 下图是一个国际象棋棋盘的示意图。其中的“王”每步可以在8个方向上行走1格，现在给定起点和终点，计算最少需要行走多少步？



|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 两个字符串分别表示起始坐标和结束坐标，字母一定是大写 |
| 输出要求 | 返回一个整数 |
| 其它要求 | 将代码写入函数func2。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| "A2","H8" | 7 |
| "E8","A1" | 7 |

1. 给定一个正整数，如果是偶数则返回各位数字中的最大值；如果是奇数则返回各位数字中的最小值。举例：给定正整数9982，该偶数各位数字中最大的是9，因此返回数字9；给定正整数1007，该奇数各位数字中最小的是0，因此返回数字0。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 给定一个正整数 |
| 输出要求 | * 如果给定偶数，则返回各位数字中的最大数字 * 如果给定奇数，则返回各位数字中的最小数字 |
| 其它要求 | 将代码写入函数func3 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| 9982 | 9 |
| 1007 | 0 |

1. 给定一个正整数，统计各位数字的因子并组成集合，对该集合中的数字进行降序排序，并以列表的形式返回排序的结果。举例：给定正整数207，数字2的因子是1与2，数字0没有因子，数字7的因子是1与7，因此对应的因子 集合（注：集合会自动去重）是{1，2，7}，排序后返回的结果为[7,2,1]。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 给定一个正整数 |
| 输出要求 | 以列表形式返回排序的结果 |
| 其它要求 | 将代码写入函数func4。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| 207 | [7, 2, 1] |
| 29 | [9,3,2,1] |

1. 给定一个数字符号构成的字符串s。将s中的奇数字符和偶数字符提取出来，分别按照原有顺序排列成两个整数。返回两个整数中较大的那个。如：输入“3052478”，可得到两个整数357和0248（即248），则返回357。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | * 字符串s长度不定。 * 如字符串s非空，则其中的字符均是数字符，且不全为“0”。 |
| 输出要求 | * 如字符串s为空，返回-1。 * 如没有偶数字符或没有奇数字符，对应的整数记为0 * 输出两个整数中较大的那个。 |
| 其它要求 | 将代码写入函数func5 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| “3052478” | 357 |
| “” | -1 |
| “1971” | 1971 |

1. 给定一个字符串列表lst。给每个字符串增加一个等长的序号前缀，并用短横（减号）连接。序号从1开始。等长序号的意思是：如果列表元素个数小于10个，则只需要1位序号；如果列表元素个数小于100个且大于等于10个，则需要2位序号；以此类推。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | * 列表lst长度未知。 |
| 输出要求 | * 如果lst为空，返回空列表。 * 列表中的空字符串不编号，且不列入输出结果中。 * 如除去空字符串后，lst中不再有字符串，则返回空列表 |
| 其它要求 | 将代码写入函数func6 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [“28”, “38”, “48”, “68”] | [“1-28”, “2-38”, “3-48”, “4-68”] |
| [“28”, “38”, “48”, “68”, “”, “12”, “12”, “4”, “126”, “8”, “12”] | [“01-28”, “02-38”, “03-48”, “04-68”, “05-12”, “06-12”, “07-4”, “08-126”, “09-8”, “10-12”] |
| [“”, “”] | [] |

1. 单词的权重定义为单词所有字母的ASCII值的平均值的整数部分。例如：“They”的权重为: (ord('T')+ord('h')+ord('e')+ord('y'))/4=102.5，则单词“They”的权重为102。给定两个单词集合s1和s2，编写程序对两个单词集合中的单词根据其权重进行统计，并构建成一个统计结果字典，字典结构为：{单词权重:不同单词个数}。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 单词集合s1和s2 |
| 输出要求 | 根据单词权重分类的字典 |
| 其它要求 | 将代码写入函数func7 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| {'personally', 'posts', 'your', 'the', 'by', 'reach', 'This', 'and'}，{'Your', 'right', 'of', 'faculty', 'adverts', 'This', 'our', 'present'} | {107: 2, 102: 2, 115: 1, 108: 3, 103: 1, 113: 1, 109: 3, 106: 1, 114: 1} |
| {'move', 'personally', 'to', 'I', 'Polytechnic', 'whether', 'and', 'be', 'your', 'worldwide'},{'will', 'roles', 'faculty', 'and', 'senior', 'your', 'in', 'with'} | {73: 1, 99: 1, 109: 4, 108: 3, 102: 1, 113: 1, 111: 1, 107: 1, 104: 1, 115: 1, 110: 1} |
| {'me', 'This', 'our', 'on', 'polytechnic', 'in', 'career', 'positions', 'your'},{'valuable', 'source', 'portal', 'related', 'Your', 'positions', 'and', 'active', 'me', 'research'} | {109: 2, 107: 3, 105: 4, 102: 2, 110: 1, 106: 1, 104: 1, 115: 1, 111: 1, 114: 1} |

1. 在情报传递过程中，为了防止情报被截获，往往需要对情报用一定的方式加密，简单的加密算法虽然不足以完全避免情报被破译，但仍然能防止情报被轻易的识别。现有一种简单的加密方法：对给定的一个特殊字符串(保证字符串首末不同时出现数字)，按照如下步骤进行处理：
2. 产生一个[0,9]范围内的随机整数n，然后对字符串中的所有小写字母、大写字母和数字字符进行循环右移n个字符，其他字符保持不变。例如：n=2时，a用c替换，y用a替换，z用b替换；A用C替换，Y用A替换，Z用B替换，0用2替换，8用0替换，9用1替换。
3. 如果字符串的首末字符都不是数字字符，则将整数n转换成数字字符并随机添加到字符串前面或者后面。
4. 如果字符串的首字符是数字字符而最后一个字符不是数字，则将整数n转换成数字字符添加到字符串的首字符前面。
5. 如果字符串的末尾字符是数字字符而首字符不是数字字符，则将整数n转换成数字字符添加到字符串最后一个字符后面。

例如：字符串abc12#5, n=3，则加密操作后为：def45#8，然后将3转换为数字字符添加到8后面，完整的加密字符串为: def45#83

现有一个**加密字符串**，请编写程序**对该字符串进行解密，获得解密字符串**。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | * 一个加密字符串s |
| 输出要求 | * 返回解密字符串 |
| 其它要求 | 将代码写入函数func8。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| "3abcde12#" | "xyzab89#" |
| "ABC@abc$09#7" | "TUV@tuv$32#" |
| "abcde12#9" | "rstuv23#" |