**考试说明**

1. **考试时间：合计180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（25分，闭卷，自动阅卷，严禁使用python编程环境进行尝试）**

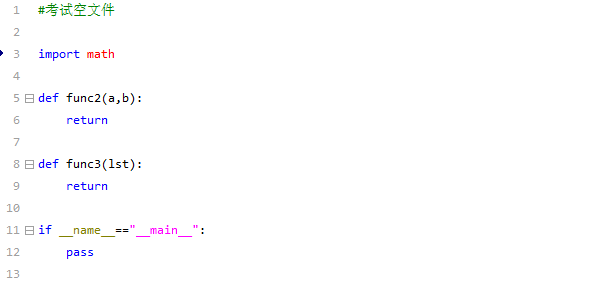
1. **打开浏览器，在地址栏中输入**http://192.168.125.3/**，点击相应链接进入登录页面。**
2. **按要求输入两遍自己的学号，考场号为所在机房号码。**
3. **点击“登录”按钮即可进入答题页面。如考试尚未开始，系统会进入等待页面并倒计时。考试开始时间到，系统会自动进入答题页面。**
4. **在页面左侧选择题号，页面右侧即会显示相应的题目。考生只需点击选择相应的选项。**
5. **答题过程中如关闭浏览器或出现系统故障导致计算机重新启动，系统不会丢失之前已经完成的题目的答案。考生可以打开浏览器重新登录并继续考试。**
6. **答题完成后，点击“交卷”按钮即可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

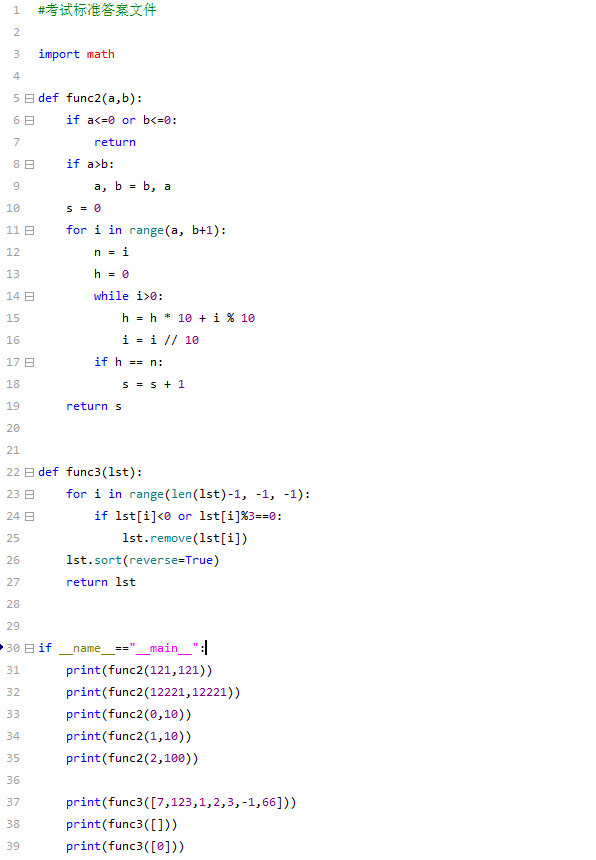
1. **下载sample.py，保存到合适的位置。**
2. **根据题目要求补全sample.py中的空函数。sample.py中的已有函数的函数名、参数数量和顺序不可以修改。**
3. **测试代码，请写入到if \_\_name\_\_=="\_\_main\_\_":部分，不要写到全局环境中。**
4. **不要在代码中调用input()，否则会导致阅卷失败。如阅卷失败，编程题部分不得分。**
5. **每个函数中不需要使用print()输出结果，而是用return返回结果。如用print()输出结果将导致该函数不得分。**
6. **不要使用关键字（if、else、break、def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、VS Code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。**



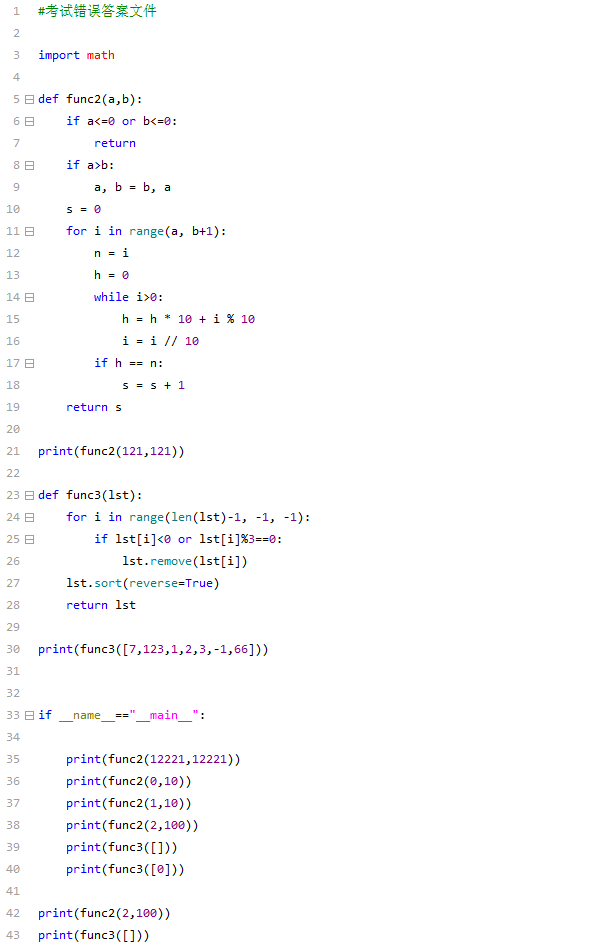
**图1 空文件样例**



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. 下图是萌蠢宠物店的服务价目表。现在给定三个参数：1) weight代表狗的体重（斤），一定是整数；2) mode，等于0时表示洗澡，等于1时表示美容；3) days表示需要寄养的天数。请根据参数计算出收费的价格。其中体重的范围为左开右闭，例如：20-30斤，表示体重大于20斤并且小于等于30斤。无论寄养多少天，至多洗澡或美容一次。



|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 三个参数都是整数，取值一定合法 |
| 输出要求 | 返回价格 |
| 其它要求 | 将代码写入函数func1。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| 8,1,0 | 80 |
| 100,0,1 | 160 |
| 30,0,3 | 200 |

1. 给定一个正整数列表，将其中奇数和偶数分别按从小到大的顺序排序。排序后的列表，原来是奇数（或偶数）的位置必须还是奇数（或偶数）。如：给定列表[2, 8, 1, 9, 7, 6]，排序后是[2, 6, 1, 7, 9, 8]，即奇数和偶数分别有序，奇偶间隔的顺序没变，依然是“偶、偶、奇、奇、奇、偶”。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 一个正整数列表，长度未知。 |
| 输出要求 | 如列表为空，则返回整数-1。  如列表不为空，则返回按要求排序后的列表。 |
| 其它要求 | 将代码写入函数func2。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [58, 90, 92, 11, 65, 76, 40] | [40, 58, 76, 11, 65, 90, 92] |
| [89] | [89] |
| [22, 92, 62, 57] | [22, 62, 92, 57] |
| [] | -1 |

1. 给定n个二元组构成的列表，每个二元组代表直角坐标系中的一个点。现需要以其中一个点为圆心，以一定的半径（整数）画圆，把所有点都包含在圆内（在圆周上也算作在圆内）。希望半径越小越好。请求出可以作为圆心的点是哪些，以及最小半径是多少。如：给定列表[(57, 13), (-44, 49), (5, 22), (56, -35), (5, 8), (27, -7)]，输出结果为(1, 3, 130)，表示序号1、3两个点可以做圆心，最小半径为130。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | n个二元组构成的列表。每个二元组的2个元素依次分别为x和y坐标值。 |
| 输出要求 | 如列表元素少于2个，则返回-1。  否则，返回一个元组。元组的元素，从小到大依次排列可以作为圆心的点在输入列表中的序号，半径作为元组最后一个元素。 |
| 其它要求 | 将代码写入函数func3。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [(57, 13), (-44, 49), (5, 22), (56, -35), (5, 8), (27, -7)] | (1, 3, 130) |
| [(57, 13), (5, 22), (56, -35), (5, 8), (27, -7)] | (1, 2, 76) |
| [(57, 13), (5, 22)] | (0, 1, 52) |
| [(57, 13)] | -1 |

1. 给定一个整数列表lst，请编程写程序删除列表中至少存在重复2次数字的整数，其它元素保持顺序不变。例如：lst=[121,12131,123,2344]，删除后的列表为: [123]

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 整数列表lst，**注意：**不保证列表元素都为正整数。 |
| 输出要求 | 返回删除特定元素的整数列表，列表元素保持原来顺序不变 |
| 其它要求 | 将代码写入函数func4。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [6967, 3411, 5893, 5369, 3175, 1537, 7311, 9847, 1221, 9716] | [5893, 5369, 3175, 1537, 9847, 9716] |
| [-9904, 5219, 9218, 6228, 3876, 9057, 8826, 7268, 4640, 8307] | [5219, 9218, 3876, 9057, 7268, 8307] |
| [] | [] |

1. 给定一个记录学生分数的字典d和一个整数n，字典结构为：{学号:(各科分数构成的元素)}，其中分数都为(0,100]范围内的正整数。编写程序找出至少存在两科成绩高于等于n的所有学生，并用学生学号构成一个升序排序的元组。

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | 记录学生分数的字典d和整数n |
| 输出要求 | 返回由至少存在两科成绩高于等于n的学生学号构成的升序排序的元组 |
| 其它要求 | 将代码写入函数func5。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| {3: (27, 77, 86, 28, 94), 14: (11, 99, 6, 4, 68), 16: (79, 78, 36, 98, 2), 40: (86, 88, 16, 60, 32)}, 80 | (3, 40) |
| {47: (94, 55, 60, 51, 91), 27: (29, 43, 8, 65, 1), 3: (93, 14, 22, 22, 17), 11: (96, 15, 65, 45, 9), 16: (43, 64, 31, 40, 42), 49: (51, 48, 12, 2, 32)}, 70 | (47,) |
| {43: (65, 62, 4, 31, 3), 21: (15, 10, 89, 24, 92), 33: (23, 59, 86, 36, 49), 24: (87, 17, 9, 57, 22), 44: (35, 7, 13, 65, 52)},95 | () |
| {},80 | () |

1. 给定一个由奇数个单词和字符#组成的字符串，计算所有单词的ASCII码值（各位字母的ASCII码值相加，区分大小写），并进行排序，最终返回ASCII值处在最中间的单词。举例：给定字符串“We##we#####Chinese######”，三个单词的ASCII码值依次是188、220、703，因此最终返回单词“we”

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | * 给定一个由奇数个单词和字符#组成的字符串; * 不会出现ASCII值相同的单词，无重复单词； * 单词区分大小写，且只包含大写或小写字母 |
| 输出要求 | * 返回排序后ASCII值处在最中间的单词 |
| 其它要求 | 将代码写入函数func6 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| “We##we#####Chinese######” | “we” |
| “#WE##LOVE###CHINA##########” | “LOVE” |

1. 给定一个字符串strn(只包含大小写字母和数字1到9)和两个正整数num1与num2，从字符串中提取在[num1,num2]闭区间范围内的连续数字，并以列表形式返回[最小值，最大值]。举例：给定strn=“abc233XF445H99999999H275nnn”，num1=100,num2=500，连续数字是233、445、99999999、275，在给定区间内的是233、445、275。最小值233，最大值445，因此最终返回[233,445]

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | * 给定一个只包含大小写字母和数字1到9的字符串； * 正整数num1与num2，且num1小于等于num2 |
| 输出要求 | * 如果没有符合条件的，返回None * 如果有符合条件的，以列表形式返回[最小值，最大值]，且**返回值必须都是整数**形式，不是字符串形式 |
| 其它要求 | 将代码写入函数func7。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| “xy123”, 100, 200 | [123,123] |
| “xy123aa124bb125HH”, 100, 200 | [123,125] |

1. 国际上GPA（Grade Point Average）有多种计算方式，某大学采用了4分制的分段绩点法，课程绩点（Grade Point）的计算方法参见下面的表格。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Score** | **Grade Point** | **Score** | **Grade Point** | **Score** | **Grade Point** |
| 100 | 4.0 | 86 | 3.6 | 72 | 2.5 |
| 99 | 4.0 | 85 | 3.6 | 71 | 2.4 |
| 98 | 4.0 | 84 | 3.5 | 70 | 2.3 |
| 97 | 4.0 | 83 | 3.5 | 69 | 2.2 |
| 96 | 4.0 | 82 | 3.4 | 68 | 2.1 |
| 95 | 4.0 | 81 | 3.3 | 67 | 2.0 |
| 94 | 3.9 | 80 | 3.3 | 66 | 1.8 |
| 93 | 3.9 | 79 | 3.2 | 65 | 1.7 |
| 92 | 3.9 | 78 | 3.1 | 64 | 1.6 |
| 91 | 3.8 | 77 | 3.0 | 63 | 1.4 |
| 90 | 3.8 | 76 | 2.9 | 62 | 1.3 |
| 89 | 3.8 | 75 | 2.8 | 61 | 1.1 |
| 88 | 3.7 | 74 | 2.7 | 60 | 1.0 |
| 87 | 3.7 | 73 | 2.6 | 0-59 | 0 |

GPA计算使用了如下的公式，其中Credit是学分。现在给定一个二维列表，列表的每一行是一条学生的课程成绩信息数据，描述如下：[学号,课程名称,分数,学分]。

请计算出每个学生的GPA并按照GPA从大到小排序（GPA相同时，学号小的在前），最后用列表的方式返回GPA排名最前的3位同学学号（如果不足3人，则全部返回）

**Grade Point Average (GPA) ＝ ∑(Credit×Grade Point) / ∑Credit**

|  |  |
| --- | --- |
| 相关说明 | |
| 输入条件 | * lst是一个二维列表 * 每一行的学号和课程名称是字符串 * 每一行的分数是整数（为了简化，取值一定在[80,100]区间） * 每一行的学分一定是整数 |
| 输出要求 | * 用列表的方式返回GPA排名最前的3位同学学号 * 如果不足3人，则全部返回 |
| 其它要求 | 将代码写入函数func8。 |

测试用例：

|  |  |
| --- | --- |
| 输入 | 返回 |
| [['2327408001','Python',85,3],  ['2327408002','Python', 100,3],  ['2327408003','Python', 84,3],  ['2327408001','Math',92,2],  ['2327408002','Math',80,2],  ['2327408003','Math',81,2],  ['2327408001','English',95,2],  ['2327408002','English',89,2],  ['2327408003','English',99,2]] | ['2327408001', '2327408002', '2327408003'] |
| [] | [] |
| [['2327408001','Python',75,3],  ['2327408002','Python', 100,3],  ['2327408004','Python', 84,3],  ['2327408001','Math',82,2],  ['2327408005','Math',97,2],  ['2327408003','Math',98,2],  ['2327408001','English',95,2],  ['2327408009','English',99,2],  ['2327408003','English',99,2]] | ['2327408002', '2327408003', '2327408005'] |