

程序设计训练第三周上机考试

考试时间：120 分钟

考试答案检查：所有源代码必须在考试结束前在网络学堂完成提交。考试结束后开始考试代码的现场演示，检查代码版本以此为准，迟交代码酌情扣分。

代码提交方式：在本地机器上以自己的“学号”为名称建立目录，在此目录中为每个题目建立子目录（如 1、2 ……），将答案源码拷贝到相应目录中，再将整个目录压缩成 zip 或 rar 文件，通过网络学堂上传答案。请确保上载内容正确，责任自负。

注意事项：

1. 考试使用的 Python 版本及 Django 版本不限。提交作业时在学习号目录下新建一个文本文件说明自己使用的 Python 版本以及 Django 版本。

查看 Python 版本号方法：在命令行中输入 `python --version`

查看 Django 版本号方法：在命令行中输入 `python -m django --version`

2. 为保证考试的公平性，除了 Django 以及 Django 的依赖库外，不能使用其他 Python 第三方库，只能使用其自带的标准库。

1. 简化版斗地主牌型检查 (5 分)

斗地主是一款著名的扑克牌游戏。本题需要编写一个 Python 程序，能够处理简化的斗地主规则，**实现牌的排序、牌型分析和比较**。

【扑克牌表示说明】：

在本题中，单张扑克牌以【花色+号码】的字符串形式表示（**本题不考虑大小王**）。四种花色以其英文单词大写首字母表示。即【S】表示黑桃，【H】表示红桃，【C】表示梅花，【D】表示方片。例如，红桃 8 表示为【H8】，黑桃 K 表示为【SK】，梅花 10 表示为【C10】，方片 A 表示为【DA】。

【简化版斗地主规则说明】：

本题只需要考虑以下四种牌型作为合法牌型（不考虑原版游戏其他牌型）：

- (1) **单张 (X)**：任意单张扑克牌。*比较规则：2>A>K>Q>J>10>9>...>4>3。*
- (2) **单顺 (ABCDE...)**：单张顺子，最少 5 张牌，最多 12 张牌。例如，【34567】，【789(10)JQK】。最长的顺子为【3456789(10)JQKA】。**注意：单顺不能有 2。***比较规则：张数相同的顺子才可以比较，BCDEF...>OPQRS...当且仅当 B>O。*
- (3) **三带一 (XXXZ)**：三张号码相同的牌加一张其他号码的单牌。例如，【2229】，【3335】。*比较规则：XXXZ>VVVY 当且仅当 X>V。*
- (4) **炸弹 (XXXX)**：四张号码完全一样的牌。例如【2222】，【AAAA】，【5555】。*比较规则：炸弹大于上面四种合法牌型中的任意牌。两个炸弹之间 XXXX>YYYY 当且仅当 X>Y。*

【题目要求】

编写一个 Python 文件命名为 exam01.py。程序接收命令行参数，第一个参数为功能选项：

(1) 当功能选项参数为“-sort”时，第二个参数为逗号分隔的 n 张牌 (1<=n<=52) 的字符串。程序需要将这些牌**升序**排序后直接输出在命令行中。排序规则为：两张牌首先按【3<4<...<9<10<J<Q<K<A<2】的规则比较其号码，如号码相同，再按照【黑桃<红桃<梅花<方片】的规则比较其花色。输出时格式与输入时相同，一行中的牌同样用逗号隔开。(1 分)

(2) 当功能选项参数为“-check”时，第二个参数为逗号分隔的 n 张牌 (1<=n<=12) 的字符串，程序需要判断这几张牌是否为一个合法的牌型，并输出一个整数，对应题干中该牌型的编号（单张牌输出 1，对子输出 2，依次类推）。如果为非法牌型则输出 0。(1 分)

(3) 当功能选项参数为“-cmp”时，第二和第三个参数为两部分牌的字符串，每部分牌为逗号分隔的若干张牌。程序需要判断这两部分牌之间的大小关系。如左边牌更大，输出“>”；如右边牌更大，输出“<”；其他情况（两部分牌无法比较或其中一部分为非法牌型）输出“=”。(1 分)

(4) 当功能选项参数为其他字符串（或程序执行时未带参数），输出“Error1”。当功能选项参数正确但后续参数数量错误时，输出“Error2”。(1 分)

(5) 在 (1) (2) (3) 要求上额外考虑可能非法输入的扑克牌参数。包括：a.非法的扑克牌字符（如【HD】，【12】，【abc】等）；b.不可能在一副牌中出现的扑克牌组合（如询问 2 张【黑桃 2】的牌型，比较 1 张【梅花 5】和 1 张【梅花 5】等）。两种错误情况分别输出“Error3”和“Error4”。注意，此错误判定的优先级高于非法牌型的判定（例，样例输入输出的最后一行）。并且 Error3 的优先级判定高于 Error4。(1 分)

【注意事项】：本题将使用程序自动判定，请务必保证程序名以及输入输出格式与题目要求保持一致。

【样例输入输出】

python exam01.py -sort C2,H3,DK,CK,SA,DJ	输出 : H3,DJ,CK,DK,SA,C2
python exam01.py -check SA,HA,C3,DA	输出 : 3
python exam01.py -check SA,HK,DJ,C10,S6,S2,H8,D7,D9,DQ	输出 : 0
python exam01.py -cmp H5,D5,S7,C5 H3,D3,S3,C3	输出 : <
python exam01.py -cmp H5,D5,S7,C5 S3,C4,S5,H6,D7	输出 : =
python exam01.py -cmp H5,D7,S7,C5 H3,D3,S3,C3	输出 : =
python exam01.py wrong_para blabla	输出 : Error1
python exam01.py -sort H5,D5,S7,C5 H3,D3,S3,C3	输出 : Error2
python exam01.py -sort C2,bla,DK	输出 : Error3
python exam01.py -cmp H3,D7,S7, C3 H3,D3,S3, C3	输出 : Error4

2.HTML 文件解析（5 分）

本题需要从给定的文件“origin.html”中爬取数据。文件主体是按英文首字母分割的 25 个表格数据（可以使用浏览器打开查看），其中每个表格的每一行显示了一个国家或地区的信息。按要求编写 Python 程序解析该 html 文件中的内容并保存。

(1) 将 25 个表格分别存储为 25 个 CSV 文件，使用表格对应的标题（即大写首字母）命名（例如：第一个表格命名为 A.csv）。

CSV 文件的格式为：每行存储一条实体的各个字段，不同字段之间用逗号分隔。表头写在文件的第一行。例如，A.csv 文件的前几行格式如下所示：

国家或地区,中文全称,英文简称,ISO 代码
阿布哈兹,阿布哈兹共和国,Abkhazia,无
阿富汗,阿富汗伊斯兰共和国,Afghanistan,AF-AFG-004
阿尔巴尼亚,阿尔巴尼亚共和国,Albania,AL-ALB-008

(2) 将“origin.html”中的 25 个表格合并为 1 个表格并存储为 all.html，只保留 1 个表头。注意每行数据的格式不要变（保留国旗图片、超链接、行背景颜色等）。

(3) 删除“origin.html”中所有带颜色的行（不包括表头）及其内容，并在最后添加为一个新的表格（表格前的标题命名为“#”）并删除新表格中的行的背景颜色、旗帜图片和超链接。新文件存储为 color.html。新表格的格式如下图所示。

#

国家或地区	中文全称	英文简称	ISO代码
阿布哈兹	阿布哈兹共和国	Abkhazia	无
阿尔扎赫	阿尔扎赫共和国	Artsakh	无
库克群岛 (新西兰)	库克群岛	Cook Islands	CK-COK-184
顿涅茨克	顿涅茨克人民共和国	Donetsk	
科索沃	科索沃共和国	Kosovo	
卢甘斯克	卢甘斯克人民共和国	Luhansk	
纽埃 (新西兰)	纽埃	Niue	NU-NIU-570
北塞浦路斯	北塞浦路斯土耳其共和国	Northern Cyprus	
巴勒斯坦	巴勒斯坦国	Palestine	PS-PSE-275
德涅斯特河沿岸	德涅斯特河沿岸摩尔达维亚共和国	Pridnestrovia	
索马里兰	索马里兰共和国	Somaliland	
南奥塞梯	南奥塞梯共和国	South Ossetia	
西撒哈拉	阿拉伯撒哈拉民主共和国	Western Sahara	EH-ESH-732

【注意事项】：

不要 HARD CODING：代码中不能出现表格中的文本（国家或地区名称，英文简称，ISO 代码等）。

【评分标准】：

- 成功生成 25 个 CSV 文件并且表格内容无误。(1 分)
- 成功生成 all.html 文件并且表格文本无误。(1 分)
- all.html 文件中的数据格式（图片、超链接、行背景颜色）无误。(1 分)
- 成功生成 color.html 文件并且表格文本无误。(1 分)
- color.html 文件中的数据格式（无背景颜色、旗帜图片和超链接）无误。(1 分)

3. Web 查询系统（5 分）

按以下要求制作一个人员信息录入和查询系统。

(1) 制作人员信息录入页：该界面主要由一个表单组成，表单项包含姓名（文本输入框）、性别（下拉菜单）、出生日期（date 输入控件）、电子邮箱（文本输入框）；以及一个提交按钮。当用户输入信息后点击提交按钮后：

(1.1) 如果用户输入正确，则将该人员信息录入后台；并且在页面显示消息：“录入成功，请继续录入下一条信息”；(1 分)

(1.2) 如果用户某个表单项内容未填写，则信息不录入后台，页面显示消息：“表单项不能为空”；(1 分)

(1.3) 如果用户填写了所有表单项但【电子邮件】一项格式不正确，则信息不录入后台，页面显示消息：“电子邮件格式不正确”；(1 分)

(1.4) 在用户点击【提交】后，如果录入成功（情况 1.1），则表单项中的内容全部清空；如果录入失败（情况 1.2 和 1.3），则需要保留用户上次输入的表单内容等待用户修改。(1 分)

(2) 制作人员信息列表页：该页内显示一个表格，显示表头和所有人员信息。(1 分)

【合法的 Email 格式要求】

1. 形如 “<local-part>@<domain>”

2. <local-part>部分至少有一个字符，且只能包括以下字符：

- 大小写英文字母：a 到 z 和 A 到 Z
- 数字：0 到 9
- 下划线：_
- 连字符（减号）：-

3. domain 部分由 n 个点 (.) 分割为 n+1 个部分 (n>=0)，每部分字符串不能为空且只能包括以下字符：

- 大小写英文字母：a 到 z 和 A 到 Z
- 数字：0 到 9
- 连字符（减号）：-

并且连字符不能出现在每部分的首尾位置。

【样例】

有效的 Email 地址：

abc@efg.zzz

_a01-xyz@hello-world

无效的 Email 地址：

Abc.def.ghi (无@符号)

A@b@c.com (多个@符号)

Abc\$def@g.com (非法字符\$)

@abc (local-part 为空)

abc@ (domain 为空)

abc@def. (domain 中点分割的第 2 部分为空)

abc@abc.-def (连字符出现在 domain 中点分割的第 2 部分的首位置)

【注意事项】

(a) 除了 Django 以及 Django 的依赖库外，不能使用其他 Python 第三方库，只能使用其自带的标准库。

(b) 题目中要求的**【页面显示信息】**可以通过 JavaScript 的 alert 函数完成，也可通过在查询之后的表单页上显示一行字完成。

(c) 1.2 和 1.3 中的校验步骤必须在 Django 后台完成。使用 HTML 的 `<input required="required" />` 和 `<input type="email" />` 方式或者使用 JavaScript 在前端校验均不得分。