**深 圳 大 学 实 验 报 告**

**课程名称：­ Python程序设计**

**实验项目名称： 作业1**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 潘浩源**

**报告人： 郑彦薇 学号： 2020151022**

**实验时间： 2022/4/25~2022/4/27**

**实验报告提交时间： 2022/4/27**

**教务部制**

|  |
| --- |
| **一、实验目的：**  **利用所学基础知识与数据结构，解决生活实际问题。** |
| 1. **实验方法步骤** 2. **读题，对每个问题提出解决问题的思路** 3. **对需要编程解决的题目按照得到的思路编写源代码** |
| **三、实验过程及内容：**   1. **解题思路** 2. **信用卡号码：** 3. 利用python中将输入默认为字符串，将信用卡号码视为一串字符串进行处理。 4. 获得信用卡号码的长度，从倒数第二位开始，获取每隔一位数字。因为0\*2=0，任何数加0仍为本身，因此可以只需记录不等于0的字符。 5. 将（2）中获得的字符强制转化为整型，如果字符大于或等于10，则需获得各个数位，利用对10取余、原数除以10的操作即可实现。其它小于10的数直接相加即可，将结果存放在sum1中。 6. 从倒数第一位开始，每隔一位获取没有乘以2的字符并强制转化为整型，相加，将结果存放在sum2中。 7. sum1与sum2相加，将结果赋值给sum3，sum3对10取余，如果结果为0，说明卡号有效。进一步判断该信用卡由哪家公司发行。 8. 根据不同公司发行的信用卡卡号的特点，设置选择语句，找到卡号对应的发行公司，进行输出即可。   **代码及细节解释如下：**  **1650897054(1)**  根据条件判断有效信用卡卡号对应的发行公司 |
| **1650897150(1)**  获得不需要进行乘2的数字，进行相加，并将结果存放在sum2中。  根据sum3对10取余的结果，判断卡号是否有效。  获得需要进行乘2的数字，将每个数位相加，并将结果存放在sum1中。  **1650897178(1)**   1. **分析名著：** 2. 存放数据：将三本名著存放在与运行代码同个文件夹里，并在python中打开。 3. 对文本文档中的标点符号、换行符及数字转换为空格，根据空格将文本文档中的单词进行分割。进一步处理单词列表，如果列表为空或者列表为换行列，丢弃这些行，并将得到的单词列表中的大写转换为小写。 4. 读取文件的每一行，按照（2）的做法获得单词列表，设置字典存放单词以及出现的次数。如果单词在字典中已经存在，则将次数加1；若单词在字典中不存在，则将该单词作为key加入字典中，然后将次数设置为1。 5. 设置元组，将字典中的键值信息复制。对元组进行降序排序，并将前20个单词以及出现次数进行输出。   **代码及细节解释如下：**  **1650902003(1)**  处理不是单词的字符，去除文件中空行或换行列  **1650902108(1)**  对单词出现次数进行统计，结果用字典存放  **1650902196(1)**  用元组存放结果，方便对结果进行降序排序。   1. **字符串“邻居”：**   **offByOne(str1, str2)函数：**   1. 根据这种“邻居”字符串的规定，首先获取两个字符串的长度，进行比较。如果长度相同，进一步判断；如果长度不相同，返回False。 2. 对于长度相同的字符串，从第一位开始依次对每一位进行比较，用一个临时变量count统计两个字符串中不同的字符数。 3. 字符串比较到最后一位后，对count值进行判断：如果count值为1即只在一个位置不同，则返回True；否则返回False。   **代码及细节解释如下：**  **1650960908(1)**  统计两个字符串有几个位置不同  判断两个字符串的长度是否相同  **offBySwap(str1, str2)函数：**   1. 因为字符串类型不能进行元素的赋值，因此在传入函数之前，首先将输入的字符串强制转换为列表结构，以便于函数中对字符串进行元素互换操作。 2. 对于传入函数的两个字符串，首先判断长度是否相等以及两个字符串是否已经相等，若两个字符串长度不等或字符串已经相等，则返回False，否则进行进一步判断。 3. 利用切片功能复制其中的一个字符串，根据该字符串的长度设置循环，进行相邻元素的互换，若在其中一轮互换中可以得到互换的结果与另一个字符串相等，则满足条件，返回True；若循环结束仍不出现相等的情况，说明不符合条件，返回False。   **代码及细节解释如下：**  **1650963140(1)1650963140(1)**  利用切片功能复制其中一个字符串  将输入的字符串强制转换为列表，方便函数内赋值给temp  **offByExtra(str1, str2)函数：**   1. 与offBySwap类似，为了能够实现对字符串单个元素的操作，将输入的字符串强制转换为列表 2. 在函数中，判断一个字符串的长度是否比另一个字符串的长度大1，如果是，继续接下来的判断，如果不是，返回False。 3. 根据字符串的长度设置循环，首先利用切片功能赋值字符串，然后依次删去长度较长的字符串中的一个元素，判断删除当前元素后的字符串是否与另一个相同，如果是，则返回True；如果不是，则返回False。   **代码及细节解释如下：**  **1650971198(1)**  利用函数pop删除指定位置的元素  判断其中的一个字符串长度是否是另一个字符串长度加1  **ListOfNeighbors(str, L)函数：**   1. 首先是根据EnglishWords文本文件获得一个从存放了这些单词的单词列表，读取文件，按照换行符对单词仅从分割，获取单词列表。 2. 根据单词列表的长度设置循环，获得列表中的每个单词，将其强制转换为列表，与输入的字符串做上述3种“邻居”单词的判断，若为其中一种情况，将当前的单词添加到列表res中。 3. 查找结束后对res列表进行输出，可以得到输入单词的所有邻居。   **代码及细节解释如下：**  **1650972758(1)**  存放结果列表  强制转换单词为列表，以便于函数中的操作  处理文本文档，获取单词列表  处理文本文档，获取单词列表   1. **MasterMind：** 2. 首先是管理员输入数字，将输入默认的字符串转换为列表，设置函数对传入的列表进行判断。若输入的列表长度不为5或有数字重复，提示错误，并重新输入。这里对数字重复的判断方法如下：根据列表长度设置双重循环，若在第i个位置之后有元素与第i个元素相同，则返回False。 3. 管理员正确输入数字后，列出游戏规则并允许玩家进行输入。对玩家输入的数字同样进行合法性判断，除了长度是否为5以及是否有元素重复外，还需判断输入是否只含数字：根据长度设置循环，依次检查每个位置，如果出现非数字，返回False，并提示玩家重新输入。 4. 玩家正确输入数字后，开始判断是否猜对了数字，并开始统计玩家猜测的次数。首先是正确位数的统计，设置双重循环，从管理员数字的第一位开始，查找玩家输入的数字中是否有与第一位数字相同的数字，如果有，正确位数加1，直到循环结束；接着是正确位置位数的统计，根据数字长度设置循环，依次检查管理员数字和玩家数字相同位置的元素是否相同，若相同，正确位置位数加1，直到循环结束。 5. 若上述求得正确位数和正确位置位数皆为数字长度即5，则说明猜对了数字，输出猜测次数等信息；若不为数字长度，则说明仍需要继续进行猜数，输出当前猜测状态信息，提示玩家再次输入数字。 6. 设置提前退出游戏标志，若用户输入EXIT，则退出游戏，并输出当前猜测次数。 7. 在进行判断时，将输入转换为列表，因此在进行当前猜测状态信息的输出时，需要将列表重新转换为字符串。   **代码及细节解释如下：**  **1650982961(1)**  对玩家输入数字的合法性的进一步判断  对管理员输入数字的合法性进行判断  **1650983133(1)**  判断玩家猜数是否正确  **1650983181(1)**  设置循环统计正确位置位数  设置双重循环统计正确位数  **1650984489(1)**  提前退出游戏  管理员输入合法之前，重复输入  玩家输入合法之前，重复输入  输出数字前将数字列表转换为字符串   1. **抓狐狸：** 2. 该游戏的实现比较简单，用基础知识并借助random库即可实现。 3. 首先是输入狐狸可以藏匿的总洞口数，并以此为范围，借助random中的randint函数随机生成狐狸的第一个位置，将玩家已尝试次数初始化为1，并提示玩家输入猜测狐狸可能藏匿的位置。 4. 判断玩家输入的位置与狐狸当前藏匿的位置是否相同，如果相同，结束游戏；如果不同，进行接下来的操作。 5. 若狐狸当天在第一个洞口，则第二天只能到第二个洞口；如果狐狸当天在最后一个洞口，则第二天只能到前一个洞口；如果狐狸当天位置在中间，则借助choice函数，随机得到狐狸第二天的位置在前一个洞口或下一个洞口；得到狐狸新位置后，允许玩家重新猜测。 6. 若玩家已达尝试次数的上限，结束游戏。   **代码及细节解释如下：**  **1651053943(1)**  得到狐狸相对于前一天，第二天藏匿的位置  随机生成狐狸第一天所在的位置   1. **运行结果** 2. **信用卡号码：**   8组输入及对应的输出：  **16508954881650895488(1)**  **1650895488(2)1650895488(3)**  **1650895488(4)1650895488(5)**  **1650895488(7)1650895488(7)**   1. **分析名著：**   运行程序，得到两个作家最常用的20个单词：  **Robert Louis Stevenson:**  1650902424(1)  **Leo Tolstoy:**  **1650902371(1)**   1. **字符串“邻居”：**   **offByOne(str1, str2)函数：**  4组输入及对应输出  **1650961078(1)1650961120(1)**  **1650961177(1)1650961198(1)**  **offBySwap(str1, str2)函数：**  4组输入及对应输出：  1650963308(1)1650963345(1)  1650963378(1)1650963402(1)  **offByExtra(str1, str2)函数：**  4组输入及对应输出：  1650971316(1)1650971346(1)  1650971373(1)1650971395(1)  **ListOfNeighbors(str, L)函数：**  4组输入及对应输出：  1650972882(1)  1650972956(1)  1650972999(1)  1650973054(1)   1. **MasterMind：**   为了显示方便，这里将最大猜测次数设置为5，下面展示3种不同情况（猜数正确、超过上限、提前退出）对应的结果：  猜数正确：  1650984022(1)  超过上限：  1650984223(1)  提前退出：  1650984532(1)   1. **抓狐狸：**   为了显示方便，这里输入允许狐狸藏匿的洞口数为10，将最大猜测次数设置为5，下面展示2种不同情况（猜数正确、超过上限）对应的结果：  猜数正确：  1651054300(1)  超过上限（即游戏失败）：  1651054424(1) |

|  |
| --- |
| **四、实验总结：**   1. **该实验是利用python中列表、元组、字典等数据结构解决实际问题** 2. **在进行应用时，应注意不同数据结构不同的定义方式以及它们具备的不同功能** |
| **指导教师批阅意见：**  **成绩评定：**  **指导教师签字：**  **年 月 日** |
| **备注：** |

**注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。**

**2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内**。