建立结构体保存成员数据项typedef struct userModel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 群名称 | 名片 | QQ号头 | QQ号尾五位 | 角色 |
| GroupName | UserName | QQNumHead | QQNumEnd | UserType |
| Char[] | Char[15] | Long=QQNum/10000 | Long=QQNum%10000 | Short={1，2，3} |

群名称最长为30汉字，或60英文字符。Utf-8中，汉字占三字节，英文字符占一字节。

群名片默认七个汉字，或14个英文字符长度，若超出，则将群名片长度翻倍。

QQ号目前最长为11位，所以将QQ号拆分进行存储。同时可以进行二维哈希。

UserType==1→群主

UserType==2→管理员

UserType==3→普通成员

1. 文件读入

根据”群成员名单.txt”,分析得到以下结论

1. 文件内“群主”一词仅出现在群成员区第一个“群主”账号下，仅间隔一个换行符
2. 最后一个群成员数据项（以下简称“数据项”）后间隔两个换行符，为js代码区。
3. 每个数据项之间间隔四个换行符
4. 每个数据项内，QQ号，均位于一对圆括号内。

**文件读入方案**

读入文件缓冲流后，截取首字节，判断是否等于0x80（utf8开头标志）若非，则提示文件编码问题。

为utf-8。然后，将文件缓冲流读入tempStr字符串中，关闭文件连接。

1. 截取字符串

将读入的文件缓冲流，存入临时字符串tempStr中，使用kmp算法查找“群主”关键词。

然后倒序找到第二个换行符，标记开始listStart（整数，标记位置）。

继续顺序kmp查找“连续两个换行符”（严格，即两个换行符，非三个中的前两个）。标记开始listEnd（整数，标记位置）。

截取listStart与listEnd之间的字符串，成为成员表memberList

1. 提取数据元素建立哈希表
2. 建立哈希表

将哈希表设计为userModel \* hashA [40][50]，分别有2000个存储空间。（hashB两个成员名单文件分别生成hashA/B）

1. 提取数据，并插入哈希表

从memberList中筛选出数据项，生成一个

type struct userModel \* userItem;

1. 先根据userItem→QQNumHead计算出行号i，再由userItem→QQNumEnd计算出列号j。得到hashA[i][j]。
2. 若发生冲突，则保持列不变，对行i进行再次散列。若行已满，即行再散列失败，则执行3），若成功则执行4）。
3. 对列j进行散列一次。执行2）。
4. 散列成功，继续散列下一个数据项。若无新项，则一张哈希表生成完毕。
5. 对两张哈希表进行比较，将比较结构存入链表。

Typedef struct resNode

|  |  |  |
| --- | --- | --- |
| A数据项指针 | B数据项指针 | 下一项指针 |
| Typedef struct userModel \* | Typedef struct userModel \* | Typedef struct resNode \* |

1. 输出
2. 按照格式要求输出txt文件
3. 输出html文件