项目说明文档

数据结构课程设计

**——**单词检索统计系统

​ 作 者 姓 名： 罗吉皓

​ 学 号： 1652792

​ 指 导 教 师： 张颖

​ 学院、 专业： 软件学院 软件工程

​ 同济大学

​ **Tongji University**

目录

1. 分析 4

1.1 项目名称 ： 单词检索统计系统 4

1.2 项目背景 4

1.3 项目功能分析 4

II. 设计 5

2.1 系统设计 5

2.1.1 系统设计说明 5

2.1.2 系统设计实现 6

III 实现 8

3.1 文本文件建立的实现 8

3.1.1 思路分析 8

3.1.2 具体实现如下： 8

3.1.3 文本文件建立截屏示例 9

3.2 文本单词汇总实现 10

3.2.1 思路分析 10

3.2.2 具体实现如下： 10

3.2.3 文本单词汇总示例 11

3.3 单词定位的实现 12

3.3.1 思路分析 12

3.3.2 具体实现如下： 14

3.2.3 文本单词汇总示例 16

IV 测试 18

4.1基本功能测试 18

4.1.1建立文件 18

4.1.2 统计单词 19

4.1.3 单词定位 20

4.2.健壮性实验 22

4.2.1 文件健壮性判断 22

4.2.2 寻找单词不存在 23

V 总结 24

VI 参考文献 24

1. 分析

*1.1* 项目名称 ： 单词检索统计系统

1.2 项目背景

​ 给定一个文本文件，要求统计给定单词在文本中出现的总次数，并检索输出某个单词出现在文本中的行号、在该行中出现的次数以及位置。

1.3 项目功能分析

本项目的设计要求可以分为三个部分实现：其一，建立一个文本文件，文件名由用户用键盘输入；其二，给定单词计数，输入一个不含空格的单词，统计输出该单词在文本中的出现次数；其三，检索给定单词，输入一个单词，检索并输出该单词所在的行号、该行中出现的次数以及在该行中的相应位置。

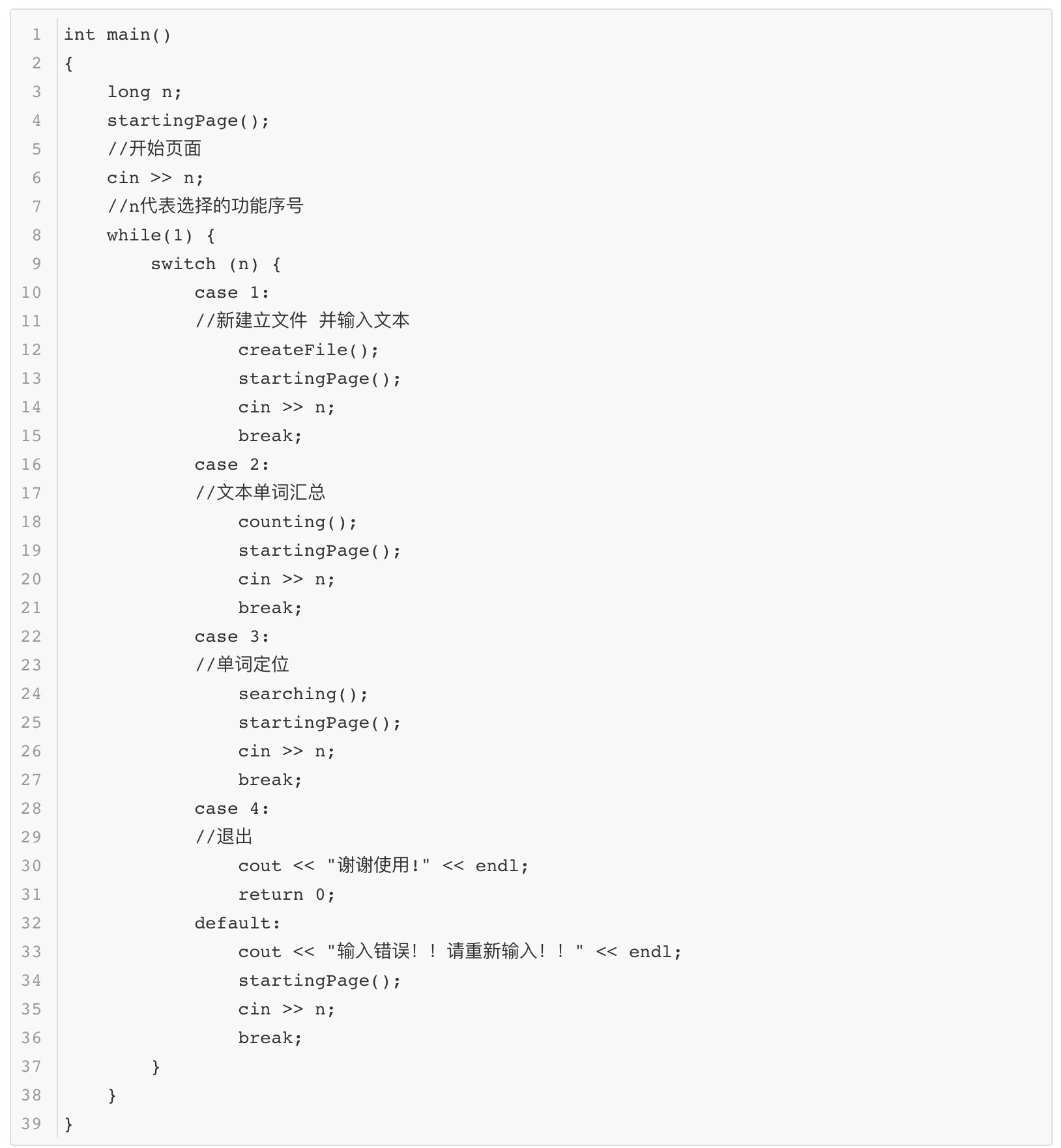
II. 设计

2.1 系统设计

2.1.1 系统设计说明

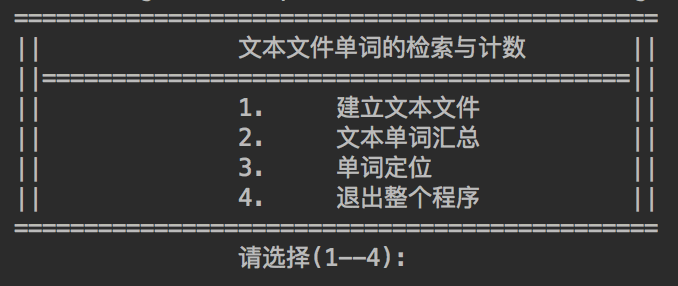
本项目主要由三个部分实现：首先通过初始化函数初始化屏幕，引导用户输入操作码进行相关操作。然后，建立一个文本文件，文件名由用户用键盘输入；给定单词计数，输入一个不含空格的单词，统计输出该单词在文本中的出现次数；最后，检索给定单词，输入一个单词，检索并输出该单词所在的行号、该行中出现的次数以及在该行中的相应位置。

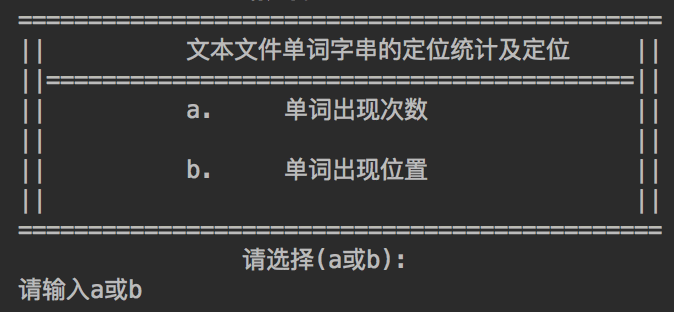
2.1.2 系统设计实现





输入功能截屏示例：



****

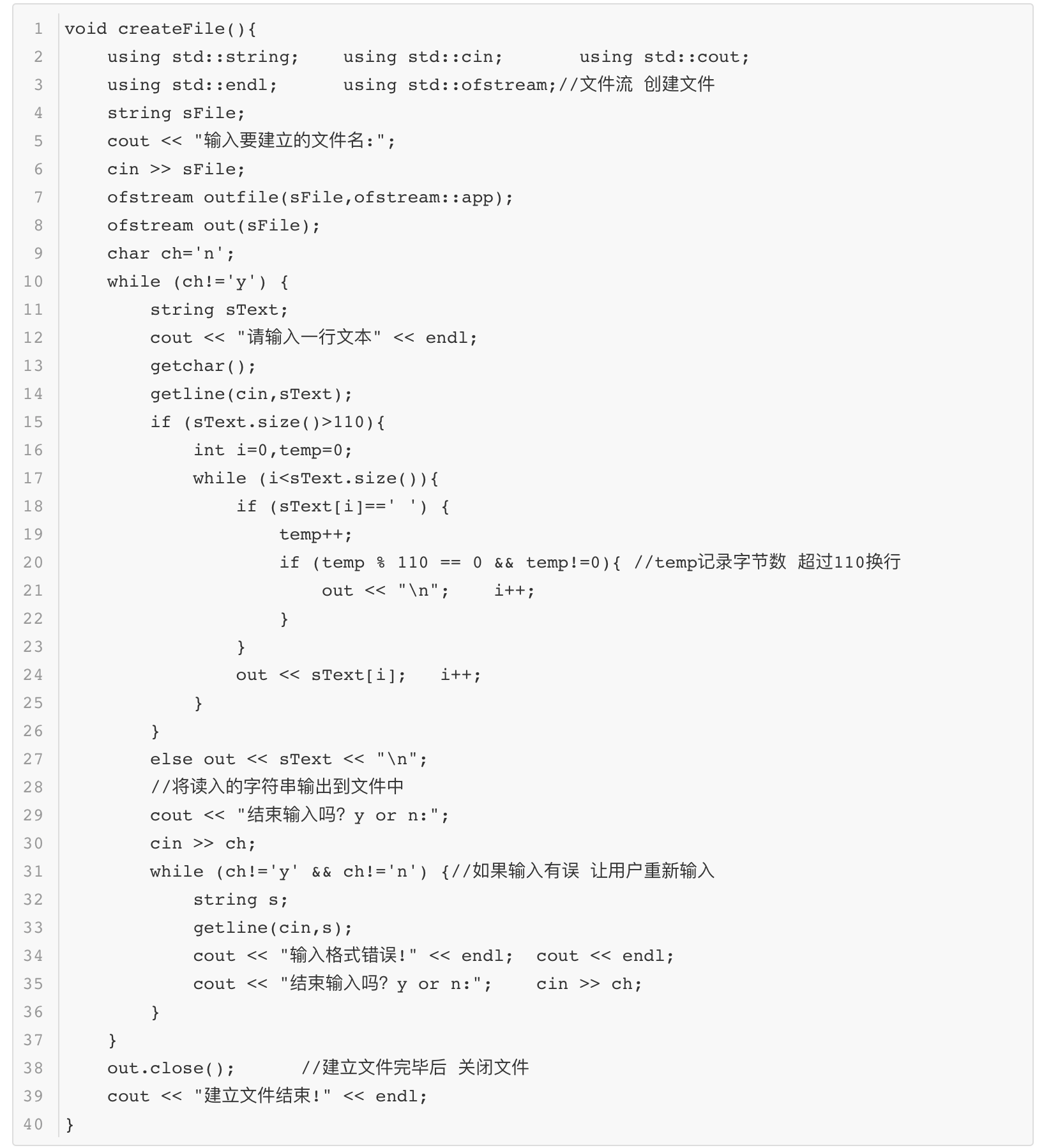
III 实现

3.1 文本文件建立的实现

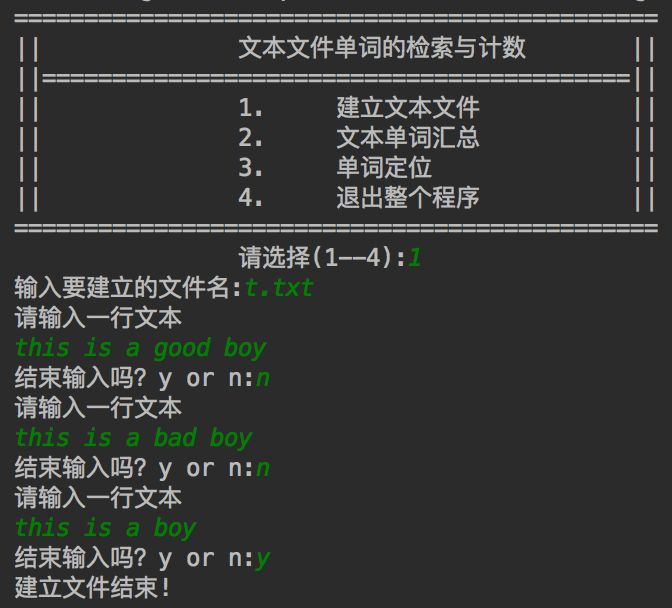
3.1.1 思路分析

使用文件流，让用户输入相关文件名，建立并打开文本文件，同过用户输入，将文本内容存入文件。

3.1.2 具体实现如下：



3.1.3 文本文件建立截屏示例



3.2 文本单词汇总实现

3.2.1 思路分析

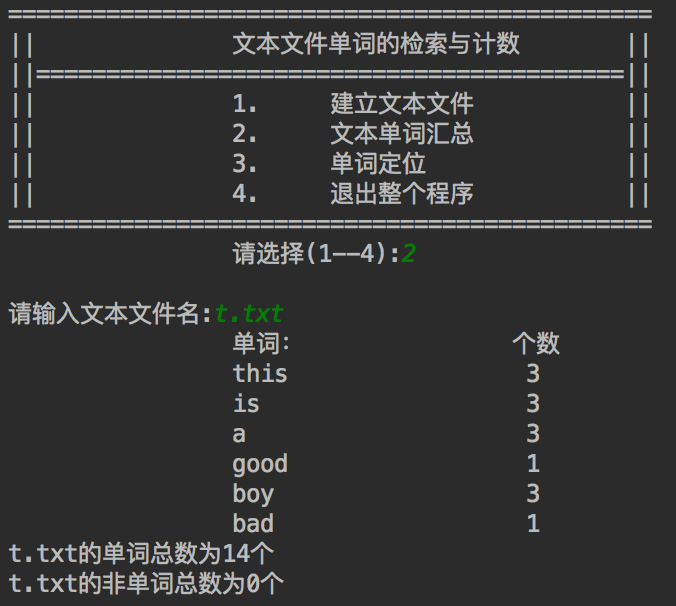
按行输入文本文件，我们通过空格判断单词的开始和结束，把单词截取出来，判断单词是否已经存在，存在的累加器加一，不存在就新建空间保存。

3.2.2 具体实现如下：





3.2.3 文本单词汇总示例



3.3 单词定位的实现

3.3.1 思路分析

在单词定位中，为了加快程序运行效率，我们使用了KMP算法。

暴力匹配的思路，假设现在文本串S匹配到 i 位置，模式串P匹配到 j 位置，则有：

• 如果当前字符匹配成功（即S[i] == P[j]），则i++，j++，继续匹配下一个字符；

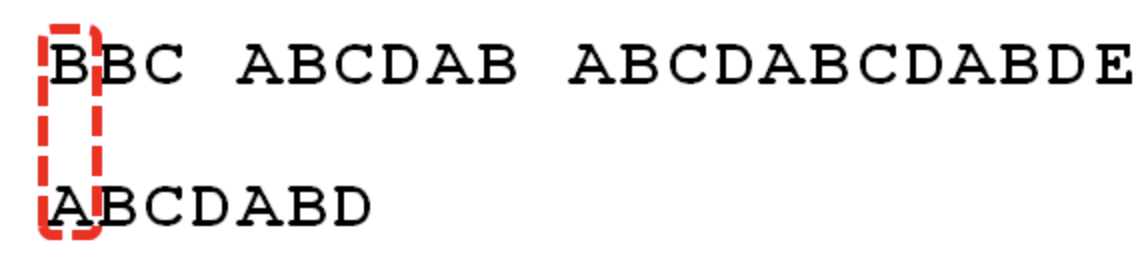
• 如果失配（即S[i]! = P[j]），令i = i - (j - 1)，j = 0。相当于每次匹配失败时，i 回溯，j 被置为0。

KMP算法概述

KMP算法用于字符串匹配问题，核心思想是找到子串中的重复出现的连续字符并将其记录到数组中。通过这种方式减少失配后回溯长度，以减少匹配次数。

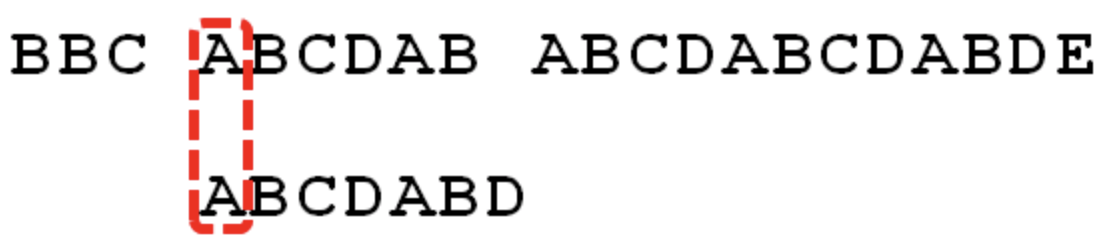
KMP算法其实是基于暴力匹配，并加上next数组之后的成果。

具体实现我们可以由下图看出：

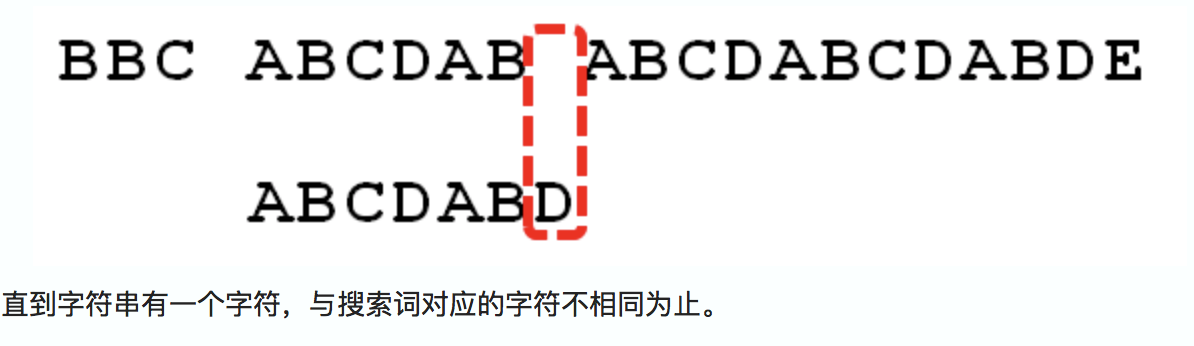


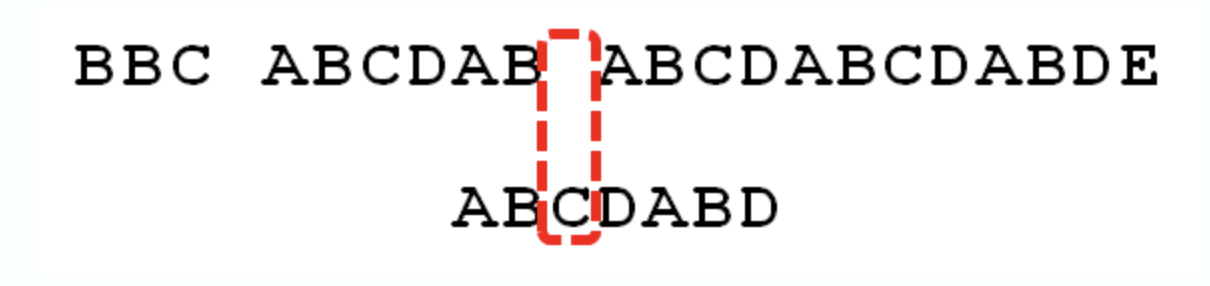
首先，字符串"BBC ABCDAB ABCDABCDABDE"的第一个字符与搜索词"ABCDABD"的第一个字符，进行比较。因为B与A不匹配，所以搜索词后移。

以此类推，直到字符串有一个字符，与搜索词的第一个字符相同为止。



继续比较





一个基本事实是，当空格与D不匹配时，你其实知道前面六个字符是"ABCDAB"。KMP算法的想法是，设法利用这个已知信息，不要把"搜索位置"移回已经比较过的位置，继续把它向后移，这样就提高了效率。

已知空格与D不匹配时，前面六个字符"ABCDAB"是匹配的。查表可知，最后一个匹配字符B对应的"部分匹配值"为2，因此按照下面的公式算出向后移动的位数：

　　移动位数 = 已匹配的字符数 - 对应的部分匹配值

　　因为 6 - 2 等于4，所以将搜索词向后移动4位。

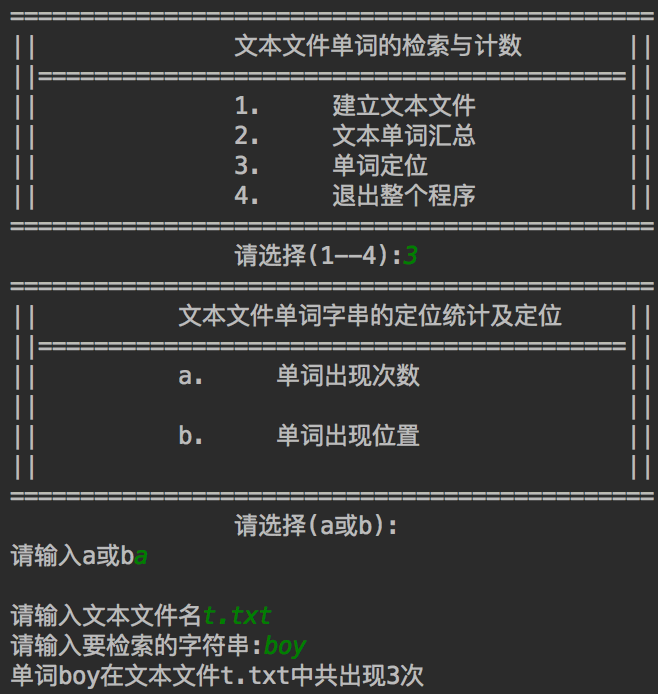
部分匹配表：

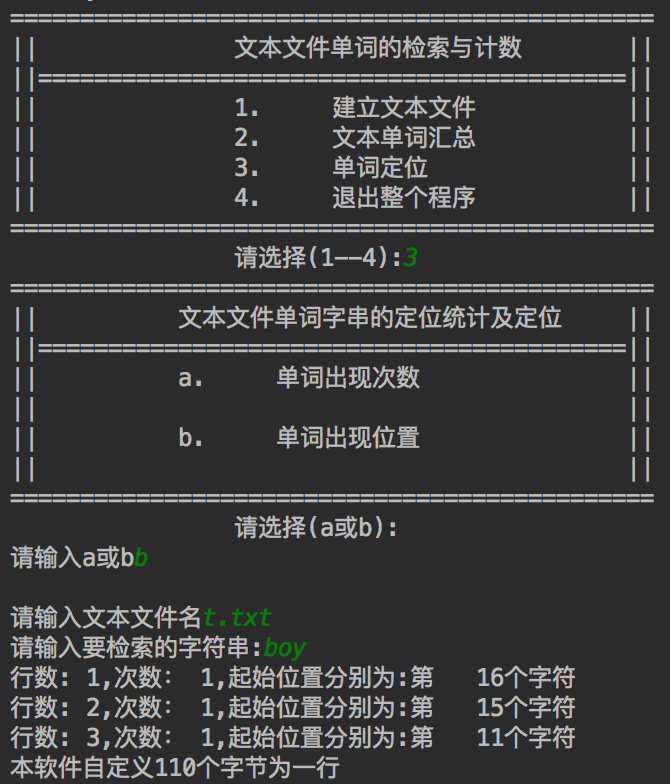
3.3.2 具体实现如下：





3.2.3 文本单词汇总示例





IV 测试

4.1基本功能测试

4.1.1建立文件

测试样例：

test.txt

Hello World

Good weather

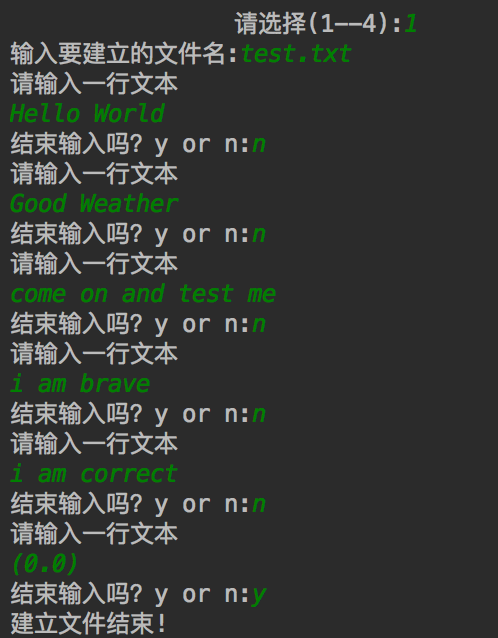
come on and test me

i am brave

i am correct

(0.0)

实际结果



4.1.2 统计单词

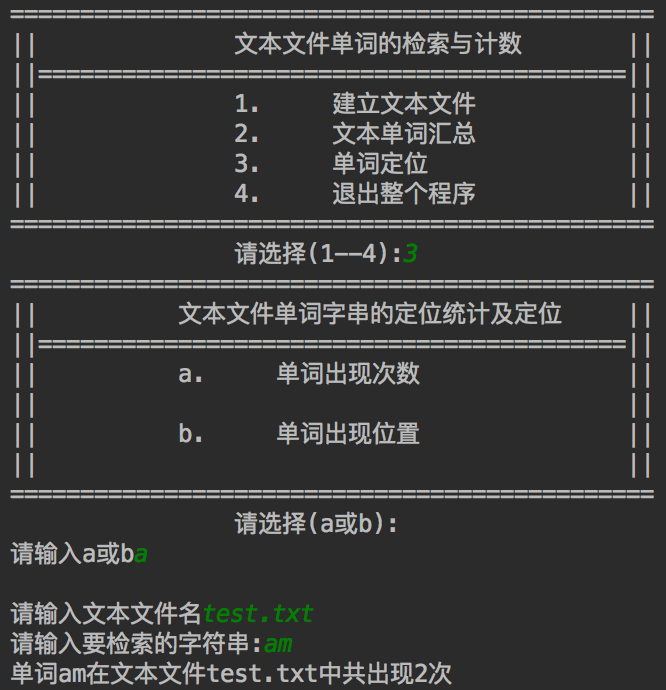


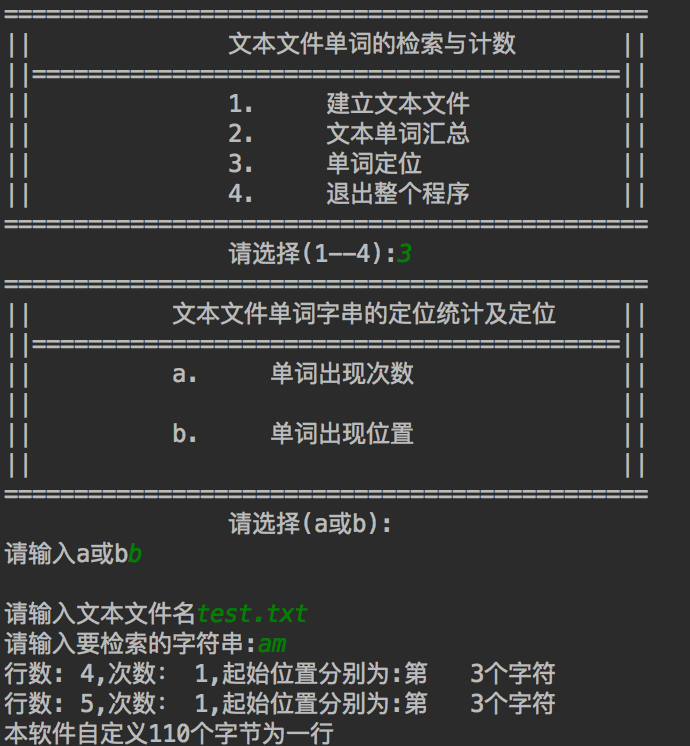
4.1.3 单词定位

测试用例

am

实际结果





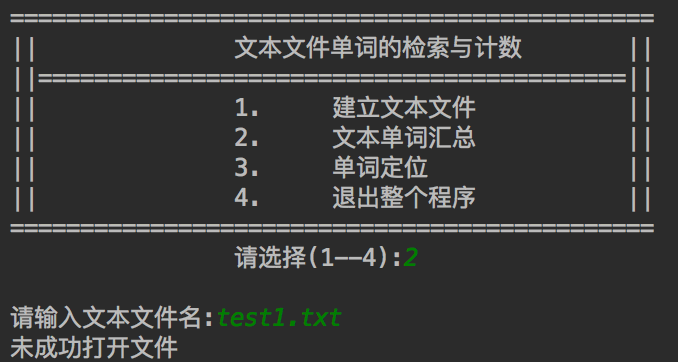
4.2.健壮性实验

4.2.1 文件健壮性判断

测试用例

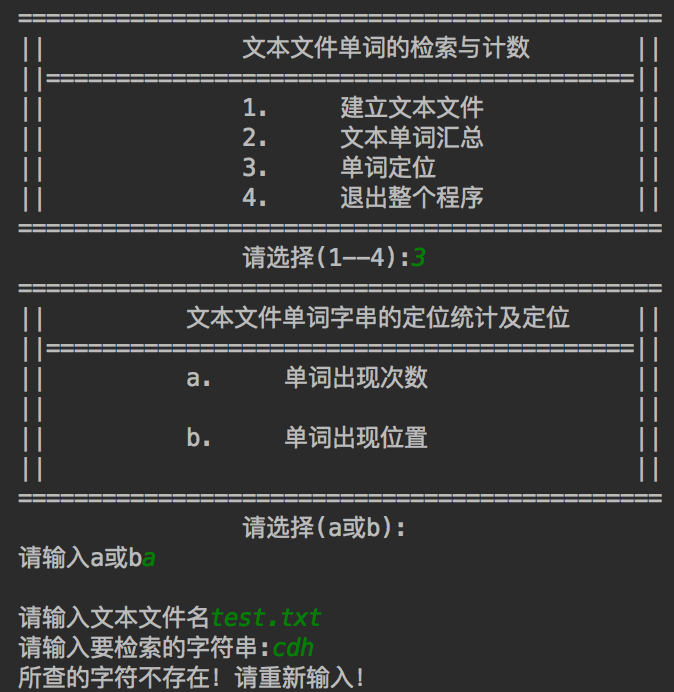
输入文件不存在

实际结果



4.2.2 寻找单词不存在

实际结果



V 总结

本次课设我近一步了解了KMP算法的应用，并熟练的掌握了文件的各种操作。

VI 参考文献

1.数据结构课本

2.<https://kb.cnblogs.com/page/176818/>