

提交方式:

Canvas 上提交

提交内容:

一个压缩包，命名方式：姓拼音_名拼音_学号.zip/rar，如：zhou_zhiming_2020000112.zip（不要用中文）
内含若干个文件夹，每个文件夹对应一道题的项目（仅保留.pro 和其他源文件；文件夹名也不要有中文）
(.cpp / .h 等)

评分标准:

代码正确性和完整性 90%

代码风格 10% （代码看着很乱的酌情扣 0-10 分。唯一的要求：代码需按层次缩进、对齐）

逾期惩罚:

逾期提交的，成绩 $\times= 0.8$ （无论逾期多久）

1. 逆反数字的和

(1) console 输入

① 一行一个字符串，表示待处理的文本文件的名称

1) 你需要从 console 读入 (cin) 这个字符串，然后打开对应文件，读取和处理文件中的内容

2) 文件中含若干行，每行一个逆反数字。具体多少行不知道，读到文件末尾。

3) 逆反数字：书写顺序从右到左，小数点写作#。如：-123.456 的逆反版本写作 654#321-

(2) console 输出

① 文件中各行逆反数字的和

2. 替换文件中的文本

(1) Console 输入

① 第一行，一个字符串 InFileName，表示待处理的文本文件的名称

② 第二行，一个字符串 A，表示欲替换掉的字符串

③ 第三行，一个字符串 B，表示欲替换成的字符串

④ 第四行，一个整数 0 或者 1，表示匹配时是否忽略大小写

⑤ 第五行，一个字符串 OutFileName，表示结果要保存到以 OutFileName 命名的文件中

(2) 任务

① 从 Console 中读取以上四个字符串，打开名为 InFileName 的文件。将其中所有的 A 替换成 B，替换后的结果保存到以 OutFileName 命名的文件中。替换时不应考虑将 A 替换成 B 后新构成的 A。例：如果文本文件中保存的内容是 aabcc，将 abc 替换成 b 的话，结果应为 abc，而不是 b

② 依次在 Console 中输出：所有发生替换的 A 在原文本中的行列信息，然后输出总替换次数

3. 文本信息统计

(1) Console 输入

① 一行一个字符串，表示待处理的文本文件的名称。可以假设文本文件中仅含字母、空格和换行

(2) Console 输出

① 前四行，每行一个信息，分别为：总行数、总单词数、总非空白字符数、总字符数（含空白字符）

② 第五行，空白行

③ 接下来若干行，每行：一个出现过的单词（按出现次数从高到低），一个整数（表示该单词出现的次数），一组行列号对（表示该单词所出现的行列位置信息，以 (i, j) 的形式表示第 i 行第 j 列）

4. 代码整理

(1) Console 输入

① 一行一个字符串，表示待处理的 c++ 代码文件的名称

(2) 任务

① 删掉代码中所有的注释，包括 // 的单行注释，以及 /* ... */ 的多行注释

② 删掉每行末尾的多余空格，如有

③ 连续多个空行的，改成仅有单一空行

④ 整理完成之后代码，保存到输入文件同一目录下，文件名的最后加上 _cleaned，后缀名保持不变