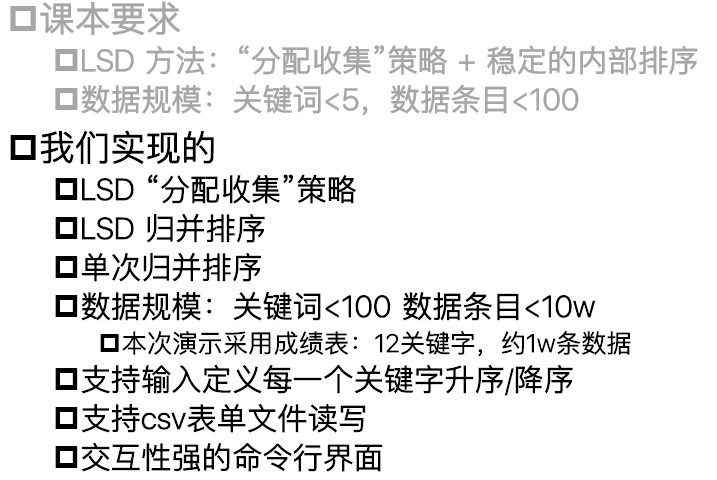
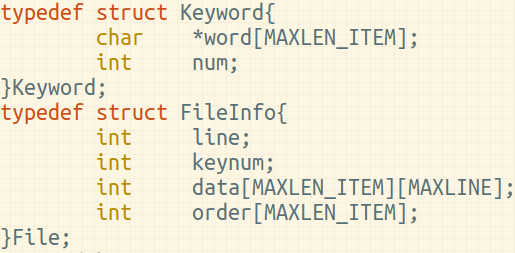
数据结构实验报告——多关键字排序

1. **本次实验我们实现了三种排序方式：**
2. LSD“分配+收集”排序：由于我们采用数组存储，故采用数组模拟桶排序的分配收集过程。
3. LSD归并排序：LSD法要求必须采用稳定的排序方法，而归并排序是稳定法里最快的
4. 单次归并排序：在比较两个条目时，比较所有关键词的值，根据每个关键词要求的升降序，直接得出大小关系。

**实现特色**



1. **存储方式简介：**





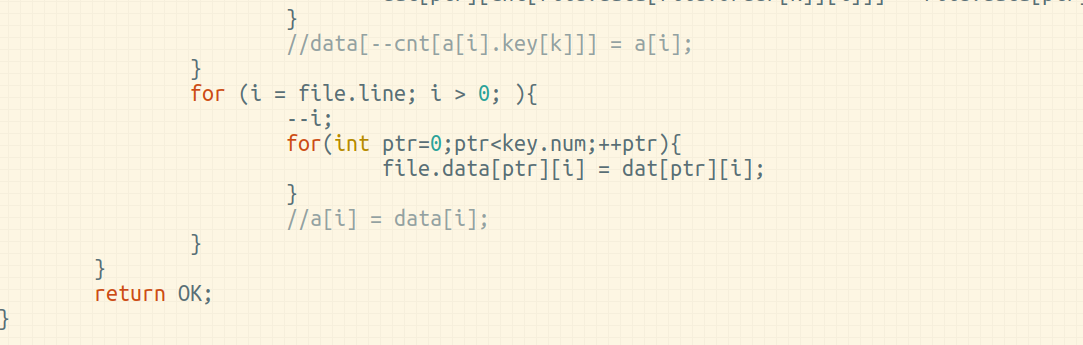
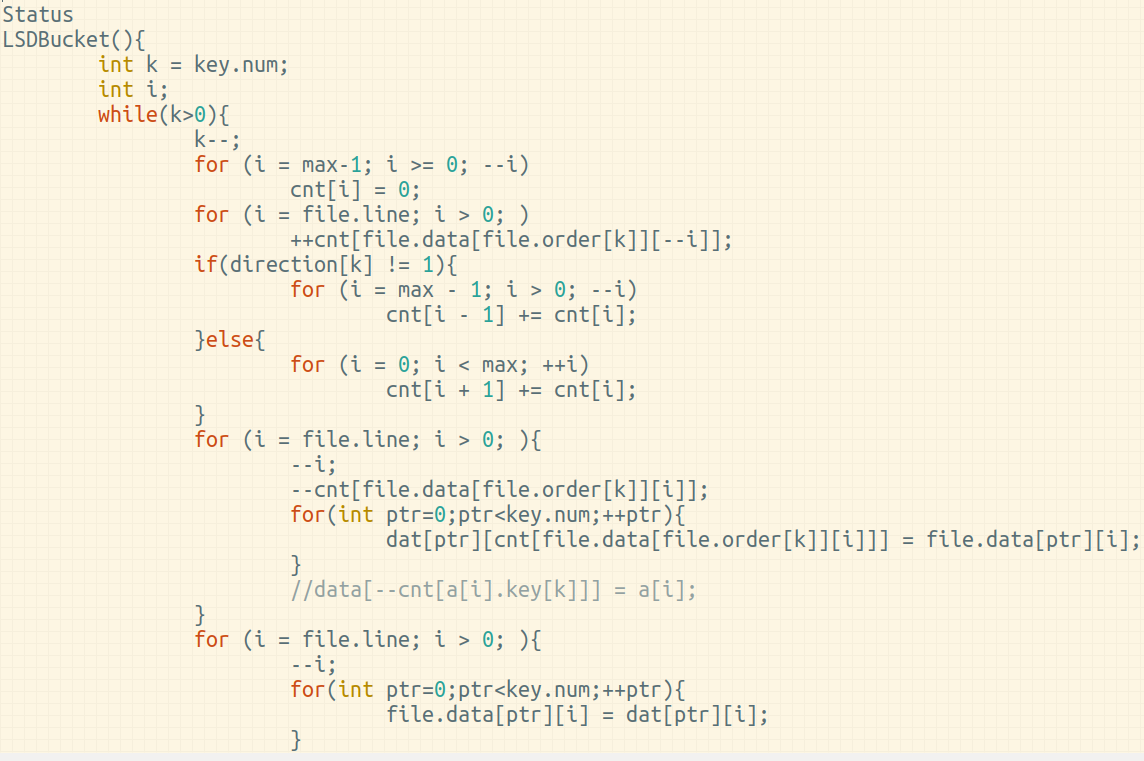
对关键词，我们采用结构体存储：num是数目，不定长字串组word是关键词

对数据，我们采用结构体存储，order是关键字的优先级

Direction表示每个关键字的升降序，1是升序，-1是降序

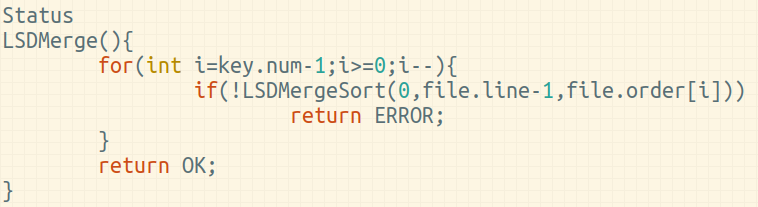
**三、 排序算法简介**

1. **LSD分配收集法（桶排序）函数：**

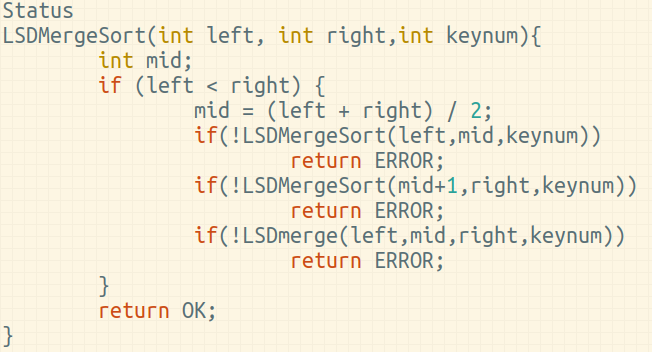


其中cnt表示其收集时应该放置的下标（这个和关键字升降序有关）。Dat数组根据数据规模分配，相当于“分配“的数据。最后我们再用一个循环”收集“起来。

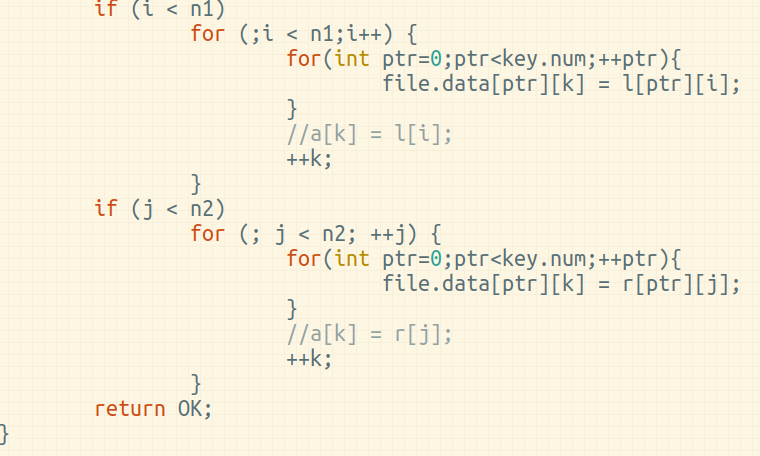
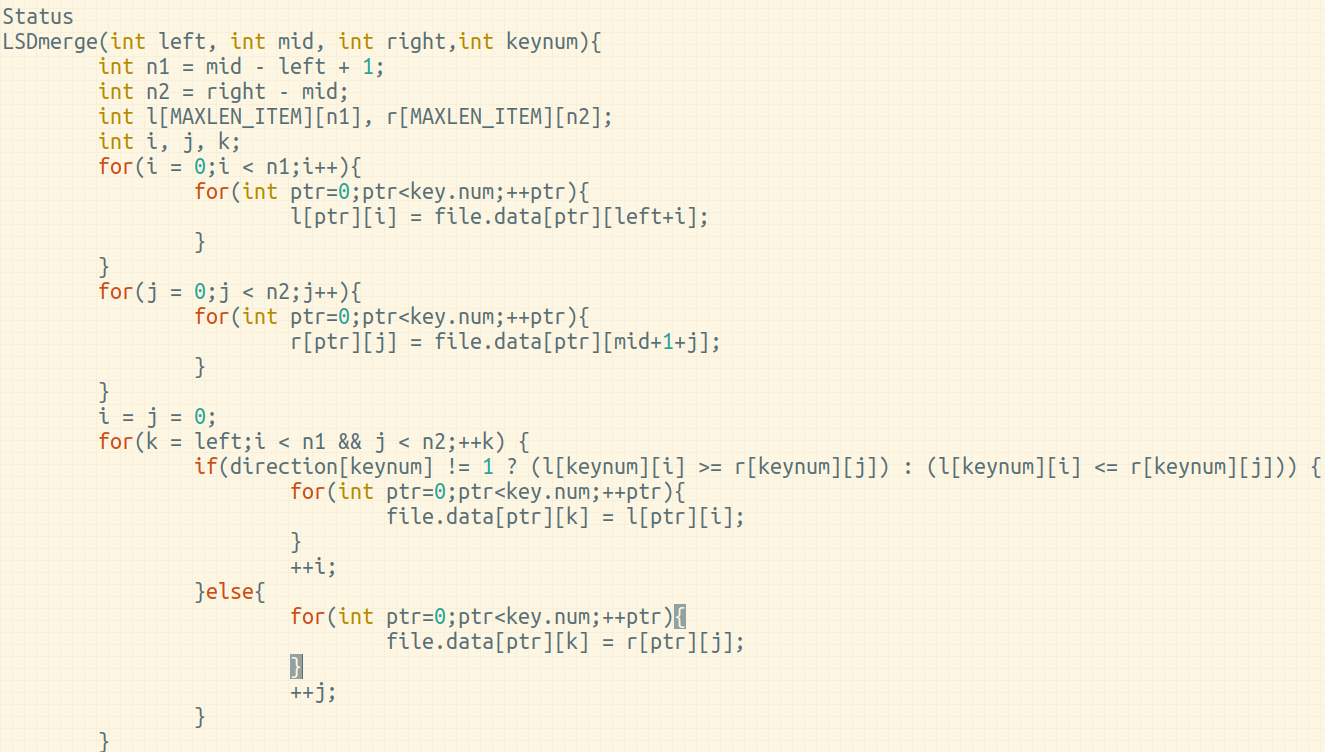
1. **LSD归并排序：**



这是主控制函数，对每个关键字从优先级低到高排序。



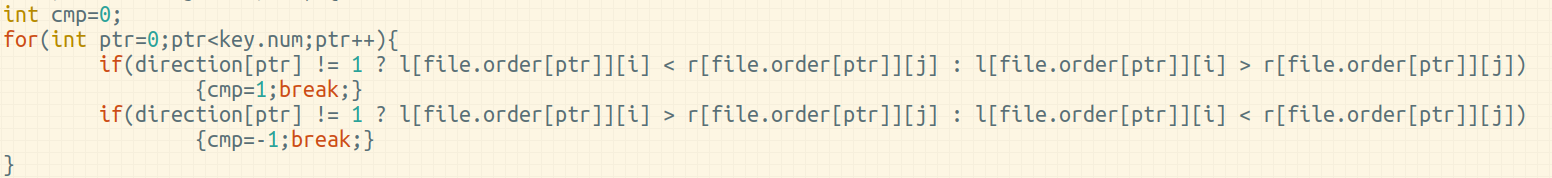
这是LSD归并“归“函数，最后调用LSDmerge再合并起来



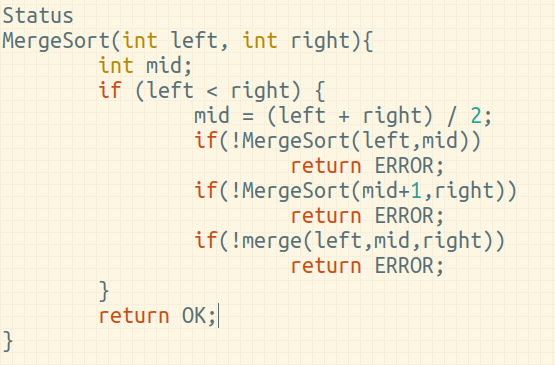
这是LSDmerge函数，与上课讲述一致。

1. **单次归并排序**

主要的变化在于：比较时我们根据优先级从高到低，并按升降序要求进行比较。这样每个条目的偏序关系是可以确定的，因此只排序一次就够了。

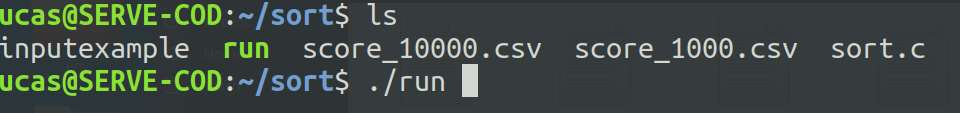


另外，我们每个函数都用Status返回值记录运行状态，方便后续调试中定位错误，如下图。

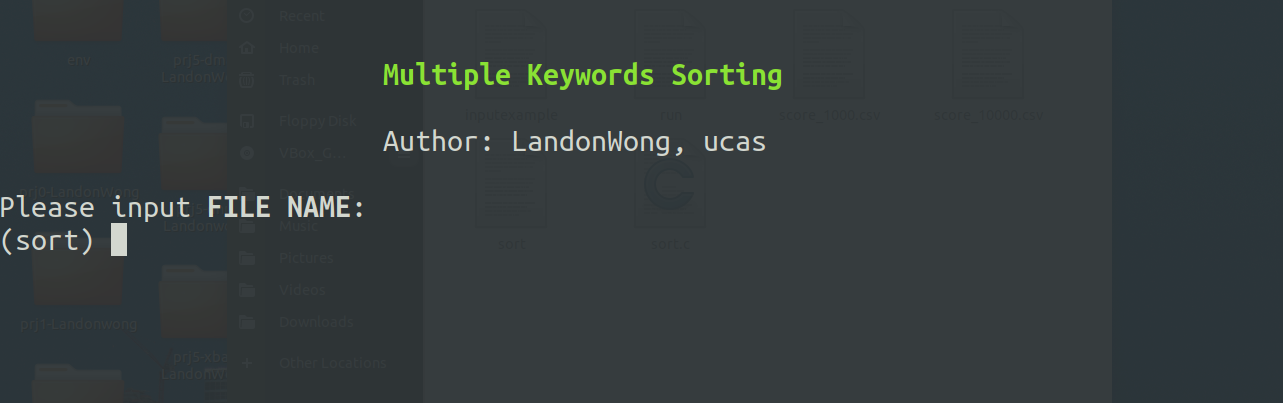


**四、 用户界面简介**

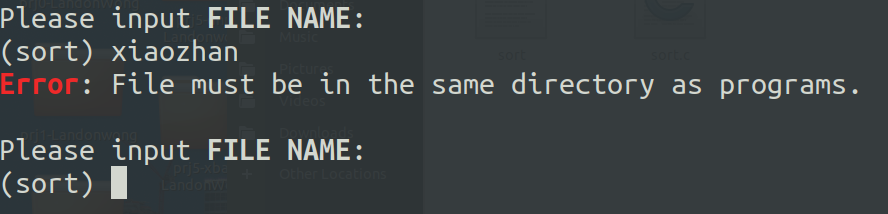
我们写好了运行脚本，每次重新编译执行



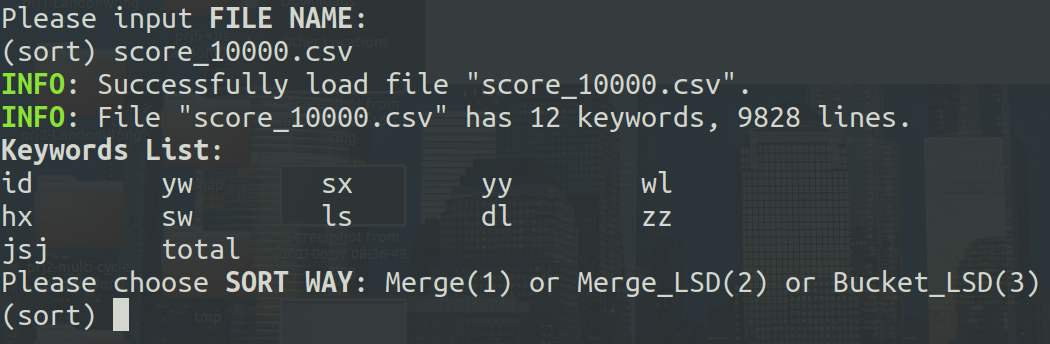
我们采用友好的命令行交互，首先提示用户读取csv文件（必须与可执行文件同目录）



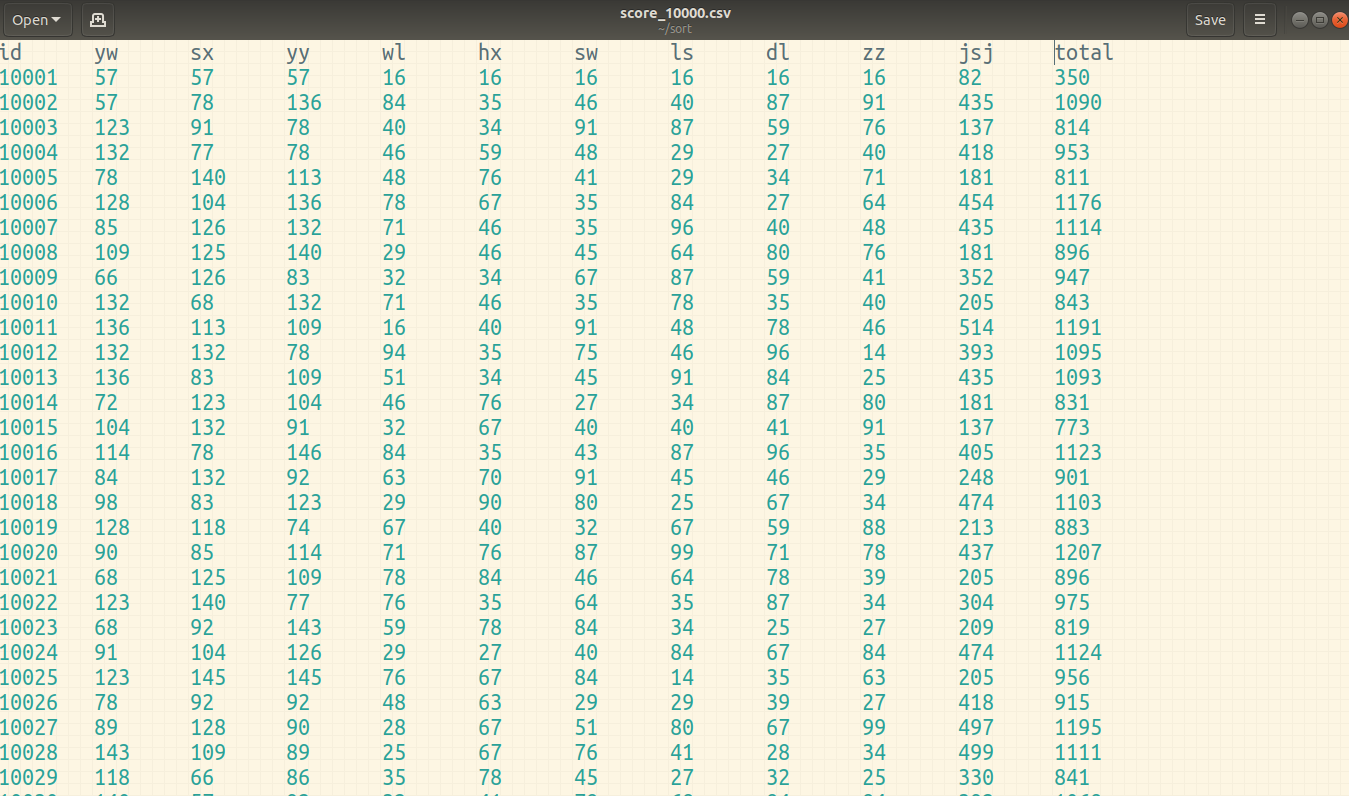
若未找到，则高亮报错，并提示重新输入



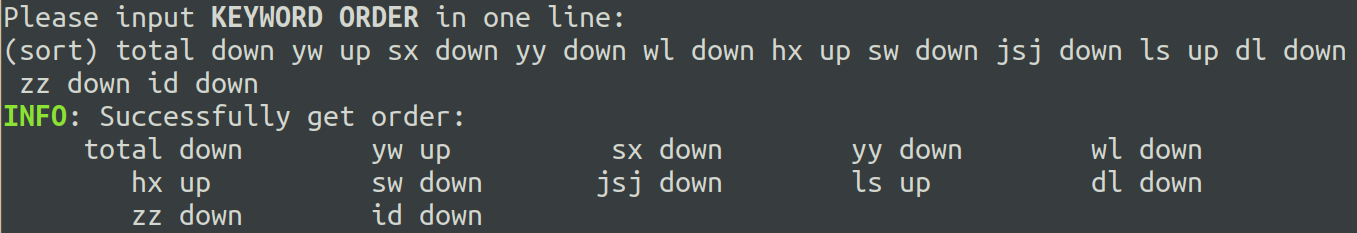
在加载文件完毕后，程序将打印关键词数目、数据条目数和关键词列表，方便用户比对并后面规定每个关键词的升降序，下面提示用户选择排序方式



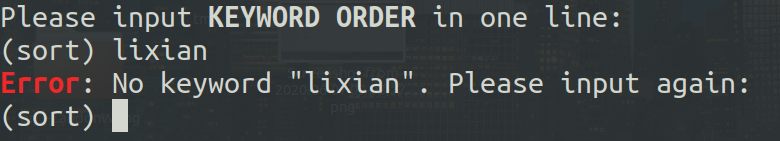
下图是我们使用的score\_10000.csv，可以看出，关键词摘取正确



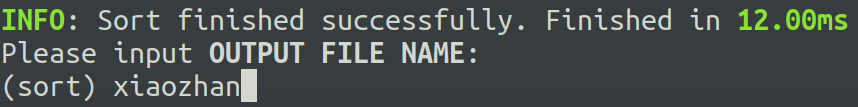
下面提示用户输入各个关键词的升降序，格式：total up（表示total关键词升序），输入完毕后将打印读取到的信息，方便用户确认（优先级：左至右，上至下）



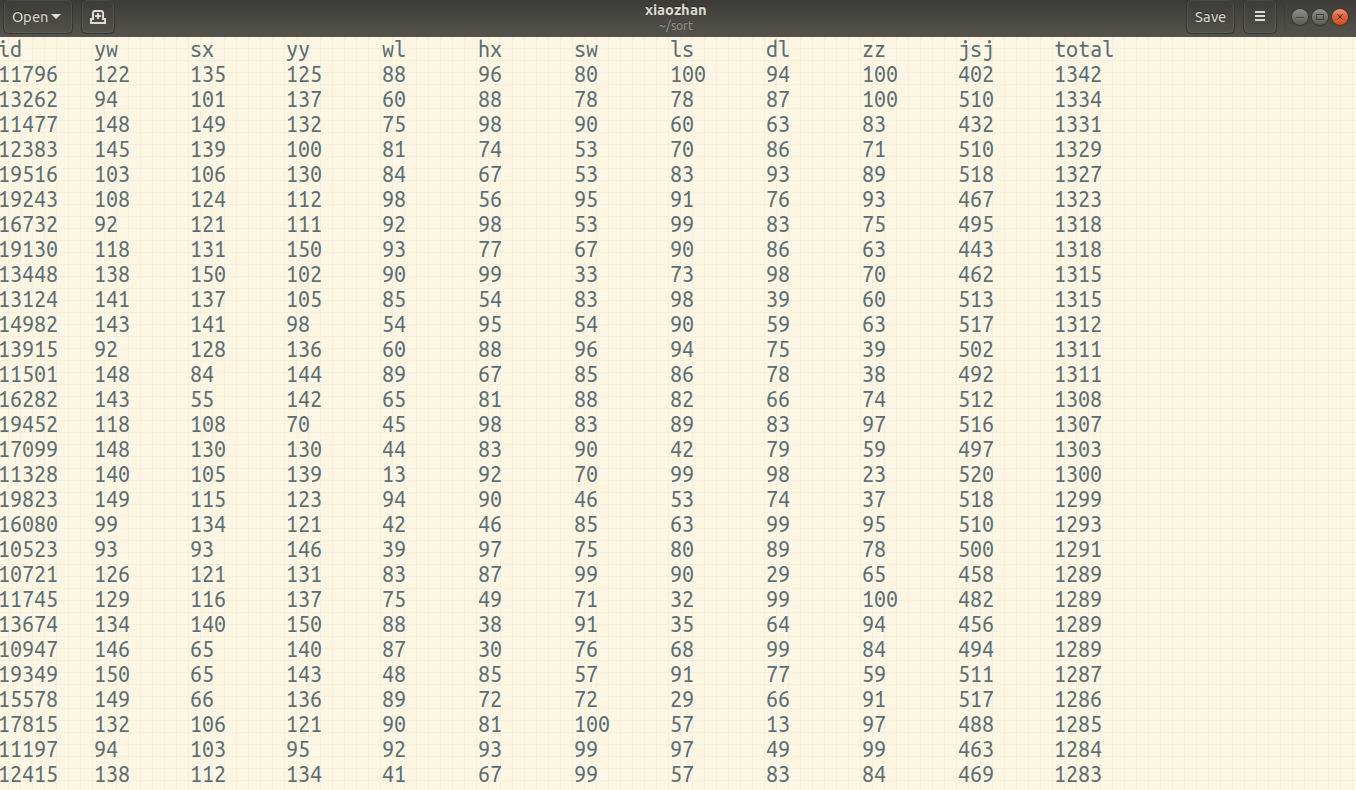
若关键词不存在，则报错，并提示从头输入



提示排序完成，打印排序消耗时间，并提示用户输入输出文件名

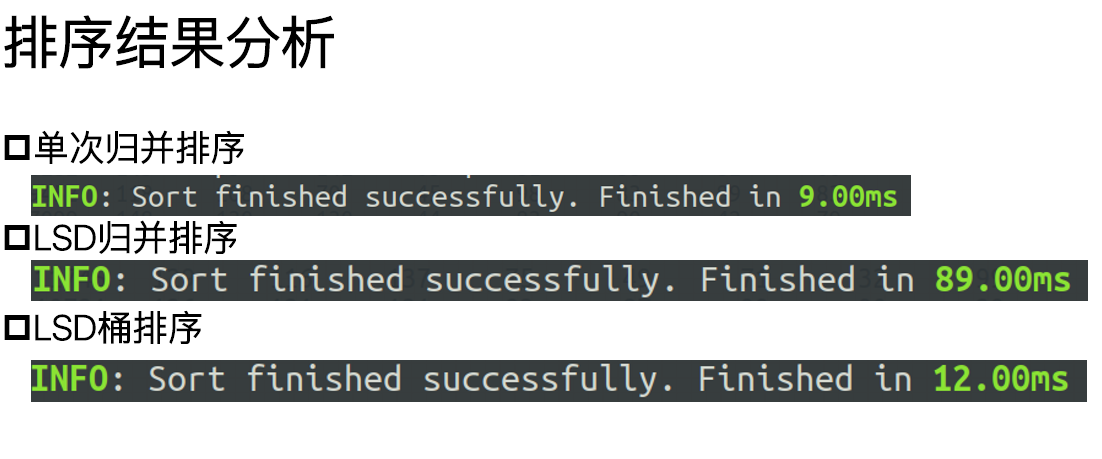


至此，交互结束，用户可打开排序完毕后的文件检查使用



可以看出，排序正确。

**五、 结果分析**

****

可以看出，桶排序拿空间换时间，即使排序关键词总量（本次测试数据是12个关键词，因此要12次）次，其效率也和单次归并排序在同一数量级。而LSD归并排序则需要进行12次归并排序，nlogn的复杂度，因此耗时增加了一个数量级。

可以看出，若采用LSD法，采用桶排序这种分配收集的基数排序更加适合。