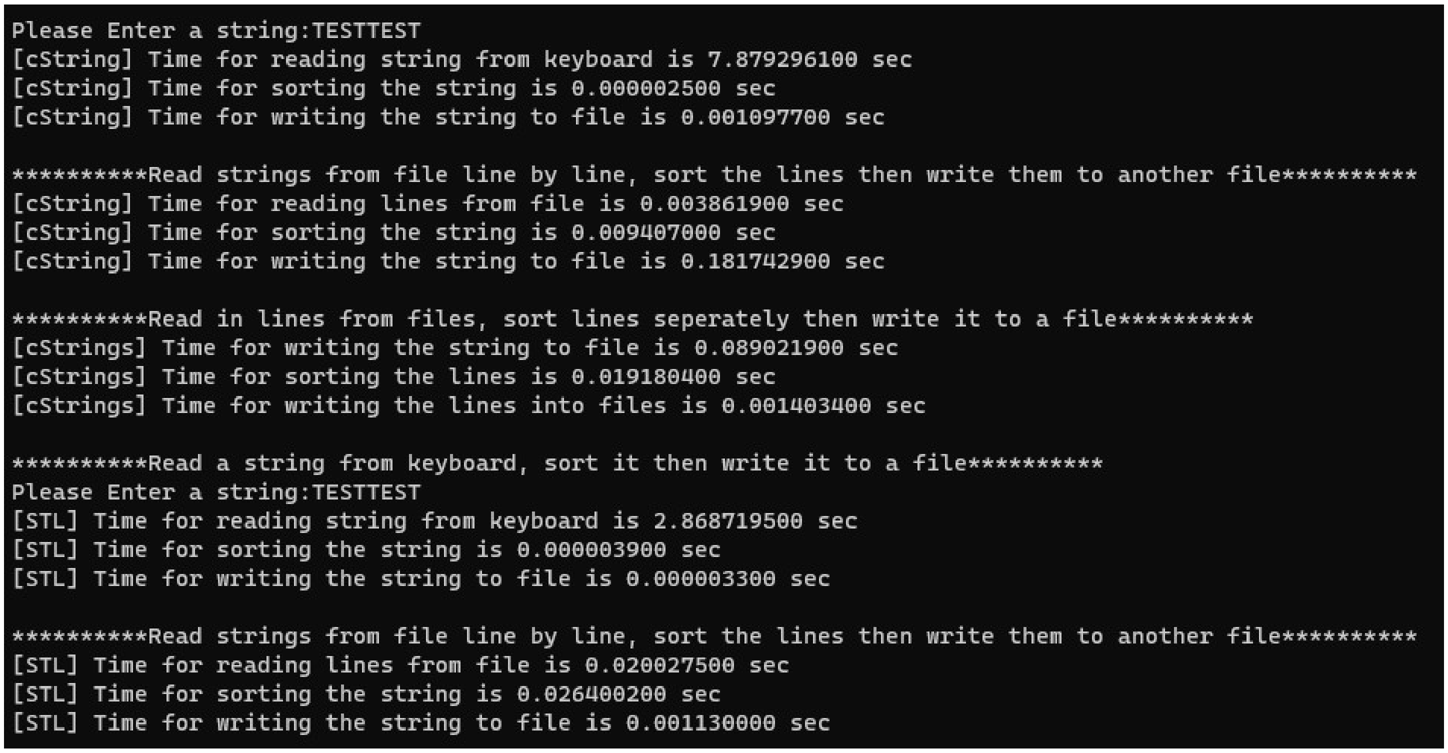
Demonstration of the task result

Speed result

The result of the task is shown below.

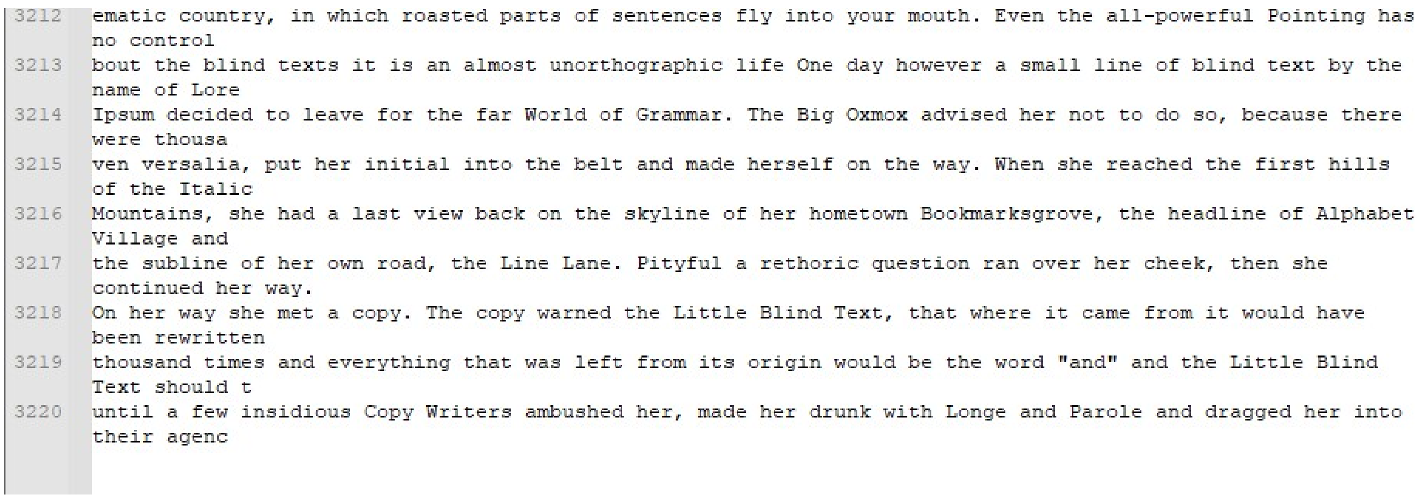


```
Please Enter a string:TESTTEST
[cString] Time for reading string from keyboard is 7.879296100 sec
[cString] Time for sorting the string is 0.000002500 sec
[cString] Time for writing the string to file is 0.001097700 sec

**********Read strings from file line by line, sort the lines then write them to another file**********
[cString] Time for reading lines from file is 0.003861900 sec
[cString] Time for sorting the string is 0.009407000 sec
[cString] Time for writing the string to file is 0.181742900 sec

**********Read in lines from files, sort lines seperately then write it to a file**********
[cStrings] Time for writing the string to file is 0.089021900 sec
[cStrings] Time for sorting the lines is 0.019180400 sec
[cStrings] Time for writing the lines into files is 0.001403400 sec

**********Read a string from keyboard, sort it then write it to a file**********
Please Enter a string:TESTTEST
[STL] Time for reading string from keyboard is 2.868719500 sec
[STL] Time for sorting the string is 0.000003900 sec
[STL] Time for writing the string to file is 0.000003300 sec

**********Read strings from file line by line, sort the lines then write them to another file**********
[STL] Time for reading lines from file is 0.020027500 sec
[STL] Time for sorting the string is 0.026400200 sec
[STL] Time for writing the string to file is 0.001130000 sec
```

- Time for reading in the string from keyboard can be ignored since it takes time for user to enters.
- From the picture, the time used by cString for sorting the lines (3220 lines) is 0.0094s which is far more faster then the c++ standard library, which takes 0.0264s. And the time used by cStrings for sorting the lines (3220 lines) is 0.0191s which is also fast than STL library. The reason for that is the programm using STL reads in the text line by line and using the container Vector to simulate a 2-d array; while the class "cStrings" use designed 2-d array. And the container vector is highly envoloped, which might be the reason why STL is slower. And for cString class, it reads in all text file as one single string, and sort the string, that's why it is more faster. The algorithm for sorting used by these 3 methods are all same, the Quick sort.

Sorting result

The text file waiting to be sorted is the file "testfile.txt". Which is shwon below:



```
ematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control
bout the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lore
Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousa
ven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic
Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and
the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way.
On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been rewritten
thousand times and everything that was left from its origin would be the word "and" and the Little Blind Text should t
until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agenc
```

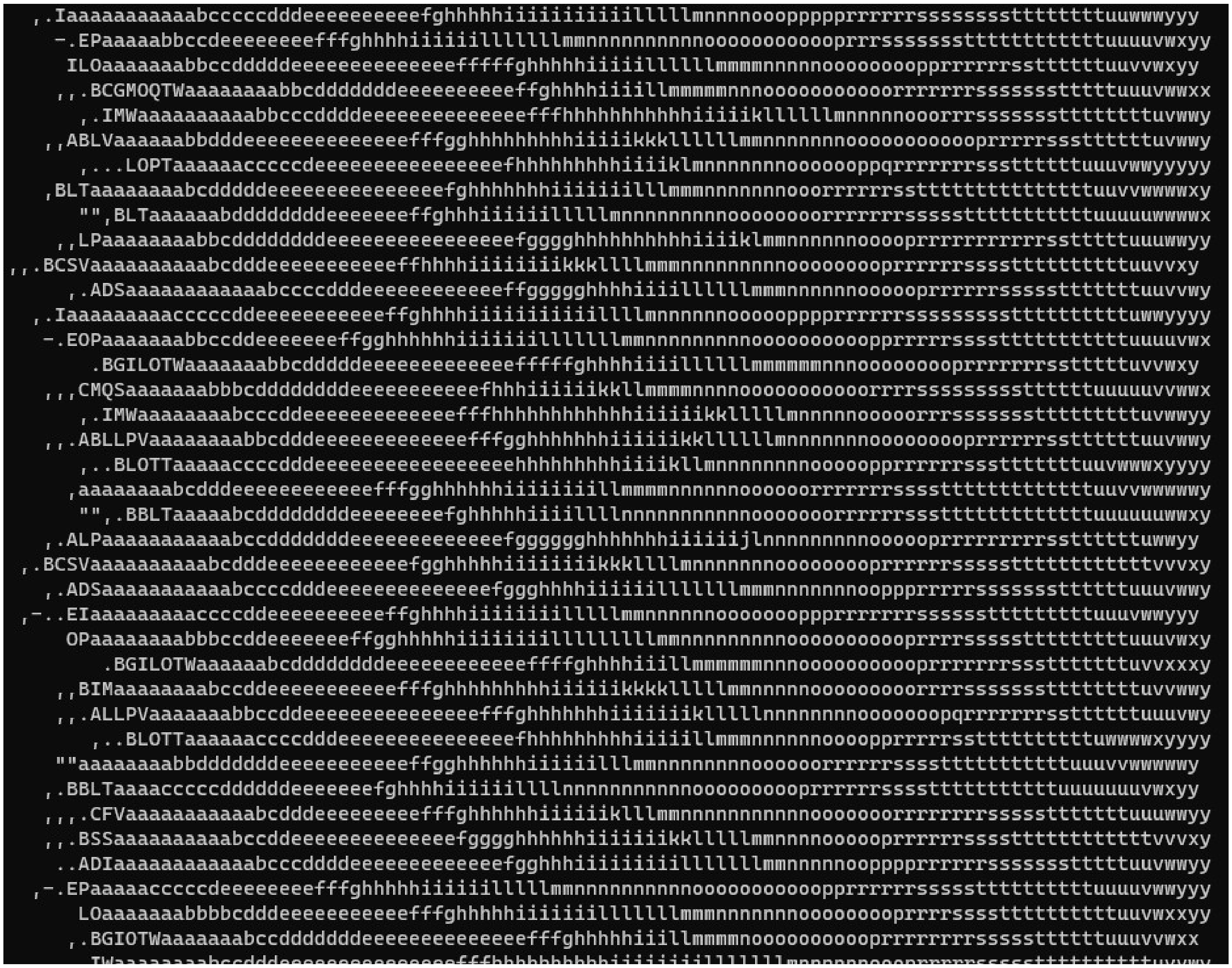
- It has 3220 lines and it has many repeted lines for the simlicity to see the result of sorting lines alphabetically.

By commenting the line 54 in the main.cpp file

```
#define _QUIET
```

One can get more output.

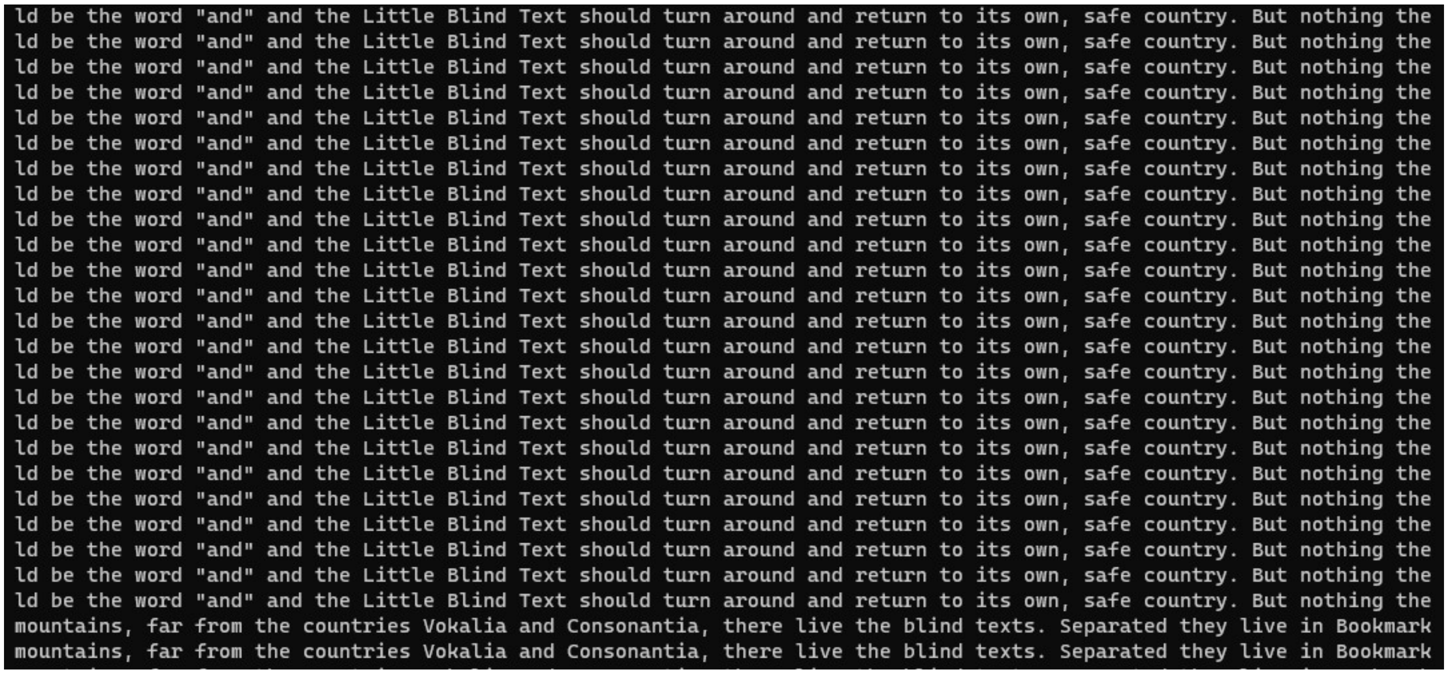
The sorting result is shown below:



```
  ,.Iaaaaaaaaaaabcccccdddeeeeeeeeeeefghhhhhiiiiiiiiiiillllllmnnnnoooppppprrrrrrssssssssttttttttuuwwwyyy
    -.EPaaaaabbccdeeeeeeeefffghhhhiiiiiilllllllmmnnnnnnnnnnooooooooooprrrsssssssttttttttttttuuuuvwxyy
     ILOaaaaaaabbccdddddeeeeeeeeeeeeeefffffghhhhhiiiiilllllllmmmmnnnnnoooooooopprrrrrrrsstttttttuuvvwxyy
    ,,.BCGMOQTWaaaaaaaabbcddddddeeeeeeeeeeffghhhhiiiillmmmmmnnnooooooooooorrrrrrrssssssstttttuuuvwwxx
      ,.IMWaaaaaaaaaaabbcccdddddeeeeeeeeeeeeeefffhhhhhhhhhhhiiiiikllllllmnnnnnooorrrssssssstttttttttuvwwy
   ,,ABLVaaaaaabbdddeeeeeeeeeeeeeefffgghhhhhhhhhhiiiiikkkllllllmmnnnnnnnnoooooooooooprrrrrsstttttttuvwwy
      ,...LOPTaaaaaacccccdeeeeeeeeeeeeeeeeefhhhhhhhhhhiiiiklmnnnnnnnnoooooooppqrrrrrrrrssstttttttuuuvwwyyyyy
   ,BLTaaaaaaaabcdddddeeeeeeeeeeeeeeefghhhhhhhhiiiiiiiilllmmmnnnnnnnnnooorrrrrrrsstttttttttttttttuuvvwwwwxy
      "",BLTaaaaaabdddddddddeeeeeeeeffghhhiiiiiiilllllmnnnnnnnnnnoooooooorrrrrrrsssssttttttttttttuuuuuwwwwx
    ,,LPaaaaaaaabbcdddddddddeeeeeeeeeeeeeeeefggggghhhhhhhhhhiiiiklmmnnnnnnnoooooprrrrrrrrrrrrssttttttuuuwwyy
,,.BCSVaaaaaaaaaaabcdddeeeeeeeeeeeeffhhhhiiiiiiiiikkklllllmmmnnnnnnnnnnnooooooooprrrrrrrssssttttttttttuuvvxy
     ,.ADSaaaaaaaaaaaaabccccdddeeeeeeeeeeeeeffgggggghhhhiiiiilllllllmmmnnnnnnnnoooooprrrrrrrsssssttttttttuuvvwy
  ,.Iaaaaaaaaaacccccddeeeeeeeeeeeeffghhhhiiiiiiiiiiiiilllllmnnnnnnnoooooppppprrrrrrrssssssssssttttttttttuwwyyyy
   -.EOPaaaaaaaabbccddeeeeeeeeffgghhhhhhiiiiiiiiilllllllmmnnnnnnnnnnooooooooooopprrrrrsssstttttttttttuuuuvwx
       .BGILOTWaaaaaaaabbcdddddeeeeeeeeeeeeeefffffghhhhhiiiilllllllmmmmmmnnnoooooooooprrrrrrrrsstttttttuvvwxy
   ,,,CMQSaaaaaaaabbbcddddddddeeeeeeeeeeeeefhhhiiiiiiikkllmmmmnnnnnoooooooooooorrrrsssssssssstttttttuuuuuvvwwx
      ,.IMWaaaaaaaaabcccddeeeeeeeeeeeeeefffhhhhhhhhhhhhhiiiiiikkllllllmnnnnnoooooorrrsssssssttttttttttuvwwyy
   ,,.ABLLPVaaaaaaaaabbcdddeeeeeeeeeeeeeefffgghhhhhhhhiiiiiiikklllllllmnnnnnnnnooooooooprrrrrrrsstttttttuuvwwy
      ,..BLOTTaaaaacccccdddeeeeeeeeeeeeeeeeeeehhhhhhhhhhiiiiikllmnnnnnnnnnoooooopprrrrrrrssstttttttuuvwwwxyyyy
     ,aaaaaaaabcdddeeeeeeeeeeeeefffgghhhhhhiiiiiiiiiillmmmmnnnnnnnooooooorrrrrrrrssssttttttttttttttuuvvwwwwwy
      "",.BBLTaaaaabcdddddddddeeeeeeeeefghhhhhiiiiillllnnnnnnnnnnnnoooooooorrrrrrrsssttttttttttttttuuuuuuwwxy
   ,.ALPaaaaaaaaaaaabccddddddddeeeeeeeeeeeeeefggggggghhhhhhhiiiiiijlnnnnnnnnnnoooooprrrrrrrrrrsstttttttuwwyy
  ,.BCSVaaaaaaaaaaabcdddeeeeeeeeeeeeefgghhhhhhiiiiiiiiiikkkllllmnnnnnnnnnoooooooooprrrrrrrssssstttttttttttttvvvxy
   ,.ADSaaaaaaaaaaaabccccdddeeeeeeeeeeeeeeefggghhhhhiiiiiilllllllmmmnnnnnnnnooppprrrrrrrssssssstttttttuuuvwwy
  ,-..EIaaaaaaaaaaccccddeeeeeeeeeeeffghhhhiiiiiiiiilllllmmnnnnnnnoooooooppprrrrrrrrssssssttttttttttuuuvwwyyy
     OPaaaaaaaaabbbccddeeeeeeeffgghhhhhiiiiiiiiiillllllllmmnnnnnnnnnnooooooooooooprrrrsssssttttttttttuuuvwxy
        .BGILOTWaaaaaaabcdddddddddeeeeeeeeeeeeeffffghhhhhiiiillmmmmmmmnnnoooooooooooprrrrrrrrsssttttttttuvvxxxy
    ,,BIMaaaaaaaaabccddeeeeeeeeeeeefffghhhhhhhhhhiiiiiiikkkklllllmmnnnnnnooooooooorrrrssssssstttttttttuvvwwy
    ,,.ALLPVaaaaaaabbccddeeeeeeeeeeeeeeeefffghhhhhhhhiiiiiiiiklllllnnnnnnnnnnoooooooopqrrrrrrrrsstttttttuuuvwy
       ,..BLOTTaaaaaacccccdddeeeeeeeeeeeeeeeeefhhhhhhhhhhiiiiiillmmmnnnnnnnnoooopprrrrrrsstttttttttttuwwwwxyyyy
    ""aaaaaaaabbddddddddeeeeeeeeeeeeffgghhhhhhiiiiiiilllmmnnnnnnnnnnoooooorrrrrrssssttttttttttttuuuvvwwwwwy
   ,.BBLTaaaacccccdddddddeeeeeeeefghhhhiiiiiiillllnnnnnnnnnnnnooooooooooprrrrrrrssssttttttttttttuuuuuuuvwxyy
   ,,,.CFVaaaaaaaaaaaabcdddeeeeeeeeeefffghhhhhhiiiiiikllllmmnnnnnnnnnnnnooooooorrrrrrrrssssstttttttuuuwwyy
   ,,.BSSaaaaaaaaaaabccddeeeeeeeeeeeeeeefggggghhhhhiiiiiiiikkllllllmmnnnnnoooooprrrrrrrssssttttttttttttvvvxy
    ..ADIaaaaaaaaaaaaabcccddddeeeeeeeeeeeeeefgghhhiiiiiiiiiillllllllmmnnnnnnooppppprrrrrrrssssssstttttuuvwwyy
  ,-.EPaaaaacccccdeeeeeeeeefffghhhhhiiiiiiilllllmmnnnnnnnnnnnnoooooooooooopprrrrrrrssssstttttttttttuuuuvwwyyy
      LOaaaaaaabbbbcdddeeeeeeeeeeeefffghhhhhiiiiiiiillllllllmmmnnnnnnnnnoooooooooprrrrssssttttttttttuuvwxxyy
     ,.BGIOTWaaaaaaabccddddddddeeeeeeeeeeeeeeefffghhhhhiiiillmmmmnooooooooooooprrrrrrrrrsssssttttttuuuvvwxx
```

- each line is sorted alphabetically

With the help of the class cStrings, lines could be sorted alphabetically without sorting the internal. The reault is shown below:



```
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
ld be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the
mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmark
mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmark
```

- it can be seen that all same lines are put together.

Out put file

Every output will be written into the file "testfilewrite.txt"

str - Yet another dynamic C string library

str is programmed originally for coping with the course "Advanced Programming" in the Hochschule Karlsruhe.

Update Logs

1. In order to cope with the task, the initial version did not use any libraries other than "cstdio" and "cstdlib". Necessary functions such as strlen, strcmp, memcpy are all self-written version. It runs very erratically and only fulfills the requirements specified by the professor's assignment.
2. The second version is mainly for writing HTTP client programs for the course "Communication and Visualization". This version fixes most bugs and comfortable use of "str" library becomes possible. It also introduces version numbers with `#define` to facilitate code interaction. From this version onwards, the "str" library starts using version number 1.0, the first version is therefore 0.9 to commemorate the learning of the HTTP protocol. At the same time, the "cstring" library was introduced and the some self-written functions were replaced with more stable version.
3. The most recent update is to cope with the second task of "Advanced Programming". In this

version, more bugs were fixed and some code even got rewritten. The "str" library can basically be used comfortably (at least by myself). The version number of this version is 1.1.

Introduction to the Data Structure

In the "str" library, the memory space for arrays is dynamically allocated by the malloc/calloc function. The metadata is the structure `str` , which contains a pointer pointing to the location of the dynamic array and a variable for storing the length of the array. Although the length of the string can be specified entirely by the variable, `size` , the `str` structure requires that the string must be terminated with a `\0` at the end. This is mainly for compatibility with the standard library "cstring". Therefore, the role of the variable, `size` , is mainly to reduce the frequency of using the `strlen` function, thus slightly improving the running speed of the program.

```
typedef struct
    {
        char* buffer;
        size_t size;
    } str;
```

During use, it is common not to define the variable `str` directly, but to define the corresponding pointers `str_p` . The advantage of using pointers is that you can manipulate values in memory directly through pointer operations, but this greatly reduces the comfort of using the program.

```
typedef str* str_p;
```

Disadvantages and further packaging

- Since using this library is basically a direct manipulation of memory using pointers, it is very inconvenient to use for those who are not familiar with the library.
- Since the library involves too many underlying operations and has too many operational pitfalls for the user, this library should be given to be further encapsulated, leaving the user with a safe interface to operate.

cString - C++ Package class for the library "str"

Requirements

- The class "cString" 1.0 version requires using the "str" library in its 1.1 version.

Introduction to data structure

As a package for the "str" library, the cString class simply has a pointer pointing to a `str` structure. The advantage of this class is that it encapsulates most of the underlying operations, which is much convenient in usability. The user does not need to use pointer operations if they do not directly manipulate the internal string.

```
private:
    str_p p;
```

Constructor

Default constructor.

```
    cString();
```

The default constructor is reserved, and its main purpose is to define a empty string.

Initialize the string by Standard input

```
cString(size_t buffer_size);
```

When using this constructor, it is necessary to specify a buffer with the size of `buffer_size` for the standard input device.

Initialize the a char string

```
cString(const char* string);
```

Initialize with a file

```
cString(const char* filename, bool lineByline, size_t lineIndex = 0);
```

By specifying `filename`, the method will automatically open the corresponding file for reading. If the boolean value `lineByline` is set to be `true`, then the method will read the specified line in the file `filename`. The line is specified by `lineIndex`, which defaults to be the first line.

Initialize with a file (User has to mangage the file handler)

```
cString(FILE* fp, bool lineByline, size_t lineIndex = 0);
```

Initialize with a str string

```
cString(str_p p);
```

cStrings - A C++ class for Two-dimensional arrays

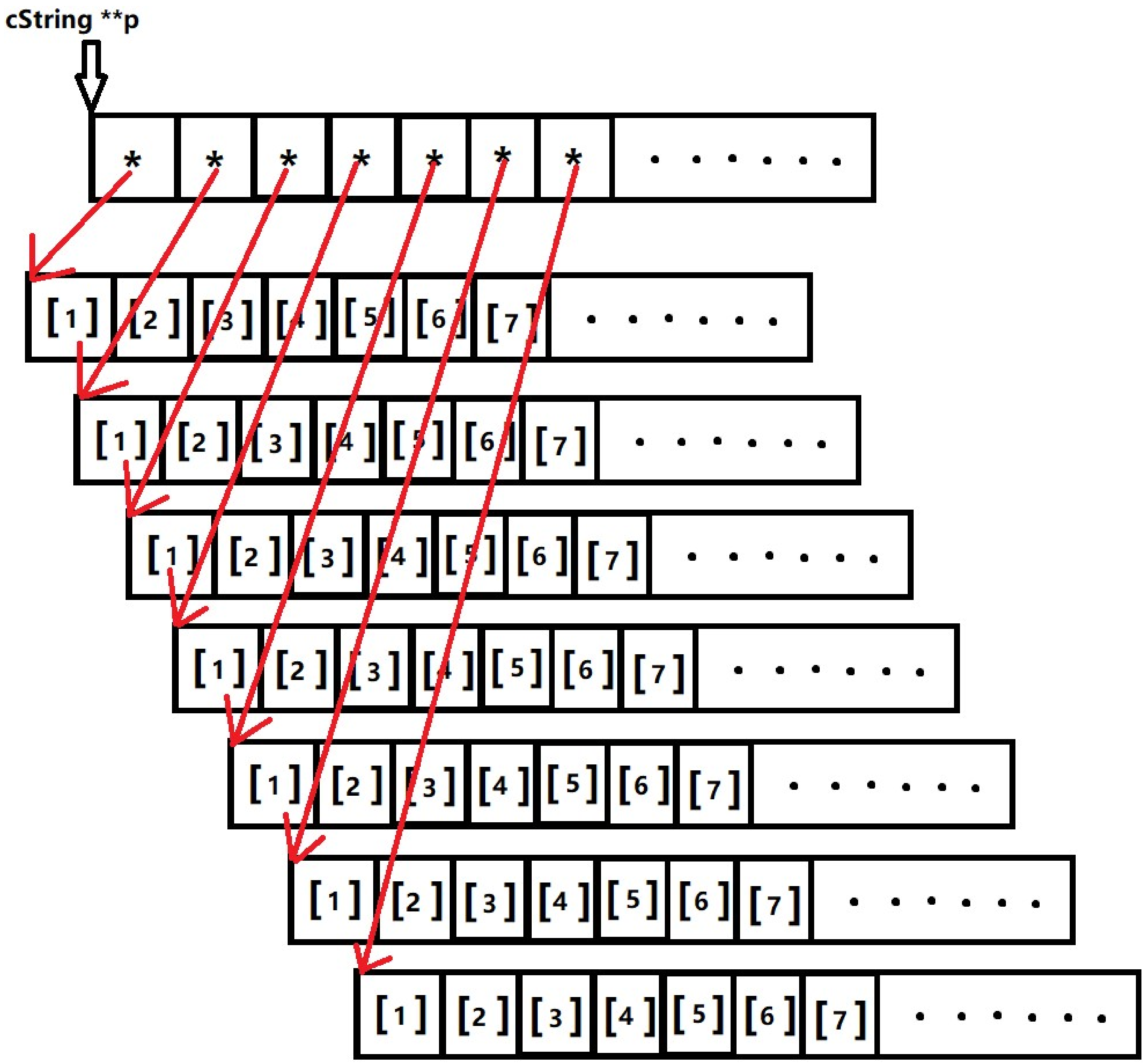
Requirements

- The class "cStrings" 1.0 version requires using the "cString" class in its 1.0 version.

Introduction to data structure

The data structure is primarily a two-dimensional array, with a two-dimensional pointer `cString **p` pointing to an array of pointers, and each pointer pointing to to an array accomodating an array of `str_p` elements.

```
private:
    cString **p;
    size_t size;
```



Constructor

The constructor of the "cStrings" class mainly constructs an array of pointers, and the differences between the different constructors are in the way how they are initialized.

```
this->p = new cString * [size];
```