

第三次实验：关键词检索

第三次实验：关键词检索

deadline

Content

Requirement

读取：建议使用 `getline` 函数进行读取

函数指针

`search_algorithm`

`vector` , `string`

运行结果：（示例）

Test Case

Input & Output

Algorithm

在前几次实验中，我们逐渐学习了 C++，当然我们也知道 C++ 是兼容 C 的。为了提高 C++ 能力，请同学们尽量使用 C++ 进行编写。我们此次提供的需要补全的代码是根据 C++ 实现的，使用 C 语言的同学请自行构造整个代码。

deadline

实验检查的 **ddl** 为 **5.8**，鉴于时间跨度较长，本次实验不接受补检查。

实验四预计于 4.24 发布

Content

实现类似 Unix 下 `grep` 命令的程序。在一个文件中查找某个关键词。（注意并不是查找到一个就结束了）

Requirement

使用 C 语言的字符串存储结构来实现字符串的操作，编写函数 `search_algorithm` 实现在一个串中查找子串的功能。然后从文件中每次读入一行，作为一个主串看待，然后查找是否存在待查找的关键词（子串），如果有则将这一行中的模式串标记为 **红色**（我们已经替你实现了，请自行阅读代码并找出指出在哪里实现的）。

读取：建议使用 `getline` 函数进行读取

使用方法

```
fstream file = fstream("./data/testcast", ios::in);
getline(file, str);
```

函数指针

为了方便同学们的检查，我们在 `search` 函数中封装了一个函数指针，详情请见示例代码

```
/**
 * @brief search the string t in s using search_algorithm
 * @param s the string to search in
 * @param t the string to search
 * @param search_algorithm the search algorithm
 * @param res the result
 */
void search(const string &, const string &, int (*search_algorithm)(const
string &, const string &, const int), vector<string> &);
```

由于我们需要使用的 `search_algorithm` 只有 `bf` 和 `kmp`，要调用 `search` 函数，只需要 `search(s, t, bf, res)` 或 `search(s, t, kmp, res)` 即可。

`search_algorithm`

我们提供的 `search_algorithm` 对应的接口为：

```
int search_algorithm(const string &s, const string &t, const int start);
```

其中 `start` 为开始查找的位置。如果查找到，返回 `t` 的第一个字母在 `s` 中对应的下标。以便于在同一个串中反复查找。如果查询失败，返回 `-1`。

`vector`, `string`

`vector<typename>` 和 `typename *var` 十分类似，请自行查询

`string` 即 `char *` 的封装

运行结果：（示例）

```
String to search: USUSTC
Result:
When I was in USTC, my dream was to study in USUSTC and get the highest GPA among USUSTCers.
But unfortunately, life played a joke on me. I was admitted to CSTC(College of Science and Technology of China).
Since then, I spared no effort to get a higher GPA. I study Quantum Mechanics day and night and finally found my place in USUSTC.
```

Test Case

我们将第一行作为模式串 t ，而后的每一行是你需要在其中查找的串 s

你可以使用我们提供的 `testcase.txt` 进行检查，也可以自行准备检查样例（我实在整不出活了）。

Input & Output

- 输入数据：从 `./data/testcase.txt` 文件中读取
- 输出数据：如上图所示，你可以使用 `printf` 或 `cout` 等

Algorithm

- BF（暴力搜索）
- KMP（串的模式匹配算法）

两个算法都需要完成。

如果你对于 kmp 算法不熟悉，可以参照这篇教程 →