

C语言实验八：文件操作

必做题要求

1. 课程内完成：OJ系统的AL~AQ题
2. 本周OJ系统截止：最好做完所有题目
3. 提交要求：提交包含AL~AQ题编号的截图，截图右上角必须包含你的学号信息

选做题要求

- 本学期所有选做题的下载和测试接口会开放到学期结束，非常建议期末考试前把它们完成并搞懂。
- 按以下要求提交，否则不计分。需提交：
 1. 相应函数实现的截图；
 2. 用相应测试例测试的运行结果截图。
- 若题目要求使用动态数组实现却未合理释放内存，则该题不计分。
- 完成奖励：
 - 9点前全部做完并检查通过后则可以提前离开课堂
 - 晚上9:15前完成并提交：每题加0.2分（加满为止，只通过部分测试例的不加分）

题目1：文件中字符串的模糊匹配

函数原型：

```
1 int FuzzyMatchLine(const char *filename, int i, const char *pattern);
```

功能描述：

编写一个函数 `fuzzyMatchLine`，实现以下功能：

1. 打开指定文件 `filename`；
2. 读取文件中的第 `i` 行（从1开始计数，最大1000行）；
3. 将该行字符串与传入的 `pattern` 进行模糊匹配（即：判断 `pattern` 是否是该行字符串的子串且不区分大小写）；
4. 若匹配成功，返回 `1`；否则返回 `0`。

⚠ 注意：

- 文件中每行仅包含一个不包含空白字符的字符串（例如："HelloWorld"、"abc123" 等）。
- 每行字符串长度不超过 100 个字符。
- 若 `i < 1` 或 `i` 超出文件实际行数时返回0。

- 若文件打开失败则返回0。
- 空字符串被认为是任意字符串的子串，因此返回1。

示例：

假设文件 `data.txt` 内容如下：

```
1 HelloWorld
2 abc123
3 ProgrammingIsFun
4 TestString
```

调用：

```
1 FuzzyMatchLine("data.txt", 2, "123");      // 返回 1 ("abc123" 包含 "123")
2 FuzzyMatchLine("data.txt", 3, "fun");      // 返回 1 ("ProgrammingIsFun" 包含 "fun")
3 FuzzyMatchLine("data.txt", 1, "world");    // 返回 1 ("HelloWorld" 包含 "world", 忽略大小写)
4 FuzzyMatchLine("data.txt", 4, "xyz");      // 返回 0
```

参数说明：

- `filename`：要打开的文件名（如 `"data.txt"`），路径为相对路径。
- `i`：要查找的行号（从1开始）。
- `pattern`：用于模糊匹配的模式字符串（不含空格，长度 ≤ 50 ）。

返回值：

- 若第 `i` 行字符串中包含 `pattern`（不区分大小写），返回 `1`；
- 否则返回 `0`。

测试代码与文件的获取方式：

```
1 wget http://10.131.21.39:8000/FuzzyMatch.c
2 wget http://10.131.21.39:8000/strings.txt
```

测试方法：

```
1 curl -X POST http://10.131.21.39:9001/judge -F "problem=fuzzy_match" -F "code=<FuzzyMatch.c"
```

题目2：学生信息文件排序

函数原型：

```
1 void LoadAndSortStudents(char mode, struct Students **stus, int *n);
```

功能描述：

定义如下学生信息结构体：

```
1 struct Student
2 {
```

```
3     char name[21];    // 姓名
4     int age;          // 年龄
5     int id;           // 学号
6 };
```

请实现一个函数，从文件读入信息，并根据传入的 `mode` 参数对 `students` 数组进行排序。

其中结构体数组的信息需要从 `stuInfo.dat` 中读出，该文件以二进制形式存储，其中前 `sizeof(int)` 个字节存储了整型数据长度 `n`，之后的 `n` 个长度为 `sizeof(struct Student)` 的字节存储了结构体的具体信息。

参数说明：

- `mode`：排序依据，取值如下：
 - `'n'`：按 **姓名 (name)** 升序排列，若姓名相同则按学号升序排列；
 - `'a'`：按 **年龄 (age)** 升序排列，若年龄相同则按学号升序排列；
 - `'i'`：按 **学号 (id)** 升序排列，学号不会冲突；

返回值：无

测试代码与文件的获取方式：

```
1 wget http://10.131.21.39:8000/SortStudents.c
2 wget http://10.131.21.39:8000/stuInfo.dat
```

测试方法：

```
1 curl -X POST http://10.131.21.39:9001/judge -F "problem=sort_students" -F "code=<SortStudents.c"
```