计算最长英语单词链(二)

把计算单词链的功能封装为独立模块，并设计单元测试

在第一阶段中，我们实现了一个命令行求解最长单词链的小程序。下面我们将逐步将我们的小程序升级为能稳定运行，给用户提供服务的软件。

大家的代码都各有特色，大家写的“软件”也有一定的用处。如果现在我们要把这个功能放到不同的环境中去（例如，命令行，Windows 图形界面程序，网页程序，手机 App），就会碰到困难：许多同学的代码都散落在各个函数中，很难把剥离出来作为一个独立的模块运行以满足不同的需求。

我们看到，不同的代码解决不同层面的问题：

- 有些是**计算数据**的（例如计算单词链）
- 有些是**控制输入**的（例如 scanf，cin，图形界面的输入字段）
- 有些是**数据可视化**的（例如 printf，cout，println，DrawText）
  有些则更为特殊，是**架构相关**的（例如 main 函数，并不是所有的程序都需要某个特定格式的 main）

这些代码的种类不同，混杂在一起对于后期的维护扩展很不友好，所以它们的组织结构就需要精心的整理和优化。

我们希望把计算单词链的功能能独立出来，成为一个独立的模块（class library, DLL, 或其它），这样的话，命令行和 GUI 的程序都能使用同一份代码。为了方便起见，我们称之为计算核心"Core 模块"，这个模块至少可以在几个地方使用：

- 命令行测试程序使用
- 在单元测试框架下使用
- 与数据可视化部分结合使用

把计算核心在单元测试框架中做过完备的测试后，我们就可以在算法层级保证了这个模块的正确性。

但我们知道软件并非只有计算核心，实际的软件是交付给最终用户的软件，除了计算核心外，还需要有一定的界面和必要的辅助功能。那么这个 Core 模块和使用它的其他模块之间是什么关系呢？它们要通过一定的 **API（Application Programming Interface）** 来和其他模块交流。这个 API 接口应该怎么设计呢？为了简单，我们可以从下面的最简单的接口开始：

```
int gen_chain(char* words[], int len, char* result[]);
```

这个函数接受三个参数，`words` 为输入的单词列表，`len` 为单词列表的长度，`result` 存放单词链，函数返回值为单词链长度。

假设我们用类 Core 封装了这个接口，我们的测试程序可以是非常简单的：

```
char* input[4] = {"END", "OF", "THE", "WORLD"};

char* result[4] = {0};

/* 调用 Core 中封装好的函数 */

int len = Core::gen_chain(input, 4, result);

Assert(len == 2);
```

当然，我们这里的判断并不充分，仅判断了单词链长度，没有判断单词链的合法性，但同学们在测试时不能这样“偷懒”。

我们要把第一阶段中实现的功能封装成独立的模块并一一进行测试，比如读取单词文本文件、输出打印等。建议大家在每一步只增量修改一个模块并做测试。这里的测试包括**新模块的单元测试**与**原功能的回归测试**。每实现一个新的功能，要保证以前运行正确的例子继续是正确的。通过这样的回归测试，可以保证自己实现的系统始终是满足预定状态约束的。（请看书中关于单元测试，回归测试的内容）在确认修改的功能正确之后再签入代码。

项目要求：

- 在第一阶段基础上增量修改，实现 `int gen_chain_word(char* words[], int len, char* result[], char head, char tail)`接口，计算**最多单词数量**的最长单词链，其中前三个参数已经在上文进行了说明，head 和 tail 分别为单词链**首字母**与**尾字母**约束（如果传入 0，表示没有约束）
- 实现 `int gen_chain_char(char* words[], int len, char* result[], char head, char tail)`接口，计算**最多字母数量**的最长单词链，参数意义同 `gen_chain_word`
- 设计上次实验中参数-n 的接口，指定单词链的单词个数要求输出所有满足要求的单词链的个数，和所有满足要求的单词链，自行设计参数，名称，按照设计好的接口在个人项目基础上增量修改

文档要求：

- 详细介绍你对于上述 Core 接口的实现，并说明如何使用这些接口。
- 选择部分单元测试代码贴在文档中，并说明测试的函数，构造测试数据的思路。
- 将单元测试得到的测试覆盖率截图，放在文档中。要求总体覆盖率到 90%以上，否则单元测试部分视作无效。

具体提交内容：

- src/ 目录下为源代码，包括接口定义的代码和单元测试的代码
- 命名为组号+功能封装+组长姓名+学号的 pdf 报告文档（如 01+功能封装+张三+PB16011001.pdf）（文档需求如上所示）

- 推荐添加 Readme 文档说明文件内容和编译/使用方法， Makefile 文件编写编译规则。建议写清楚代码注释，养成良好的编译习惯。

作业提交

- 由组长将作业上传到 FTP 服务器
  - 服务器地址 114.214.161.65
  - 用户名：student
  - 密码：student@se2019
- 服务器上有两个文件夹上午班/和下午班/，根据自己对应班级提交
- 提交打包好的压缩文件，文件名为组号+功能封装+组长姓名+学号，如 01+功能封装+张三+PB16011001
- 如果需要重新提交则在末尾加上“+VN”，N 表示重新提交的次数，如 01+功能封装+PB16011001+V1
- 截止时间 3 月 31 日晚 12 点

注：有问题可发邮件或 QQ 联系助教进行咨询。