【注意:】

1、 除明确要求外，已学过的知识中，不允许使用 goto、不允许使用全局变量

2、 本作业仅要求 VS2022 编译通过即可（“0 errors, 0 warnings”）

3、 允许使用 string 类，允许使用 vector，不允许使用其它 stl 容器

综合题 6：完成一套读配置文件的工具集

【背景描述:】

在 Windows 和 Linux 操作系统中，很多应用程序都有相应的配置文件，用来设定程序运行过程中的各个选项，配置文件的结构说明如下：

```
;这是某程序的配置文件
;2023.12.17 修订

[VideoProperties]
Title=属性设置
Title_V=10

[SpecialEffect]
Title=特效
EffectBlock=12.3 #版本
ZoomBlock=

[FaceTrack]
   Title =    人脸追踪
FaceTrackingBlock=y

# FaceTrack=3
```

```
;这是某程序的简单配置文件
;2023.12.17 修订

Title_V=10
EffectBlock=12.3 #版本
ZoomBlock=
   Title =    人脸追踪
FaceTrackingBlock=y
#FaceTrack=3
```

★ 配置文件为文本文件，支持 Windows/Linux 格式

★ 配置文件分为若干组，每组用[***]表示组名，组名各不相同

★ 每组有若干项，每项的基本格式是“项目名=值”，同组的项目名不相同，不同组可能相同

- 每个项目一行，不允许多项目一行
- 项目名为字符串，由英文/中文/数字/符号等组成，均为可见字符

★ 值的可能取值有：整数、浮点数、单字符、字符串、空

- 字符串可能为英文/中文/数字/符号等
- 字符串不含“;#=[]”等特殊含义的半角字符
- 项目名及值的前后允许有空格、tab 等，不包含在内，也不算错误(左侧例子中[FaceTrack]仍为 Title=人脸追踪)

★ 如果某行出现;或#(均为半角)，则表示该符号出现至本行尾部均为注释(左侧红色)，不需要符合语法要求，也不被读取

★ 某些配置文件，可能只有项目名，没有组名，下文中称为简单配置文件

★ 其他

- 定义一行的统一处理顺序：取出一行后，先截断;及#开始的注释，再去除前后空格/tab，剩下为有效内容
- 有效内容第一个是[，最后一个是]，就认为是组名，否则不是
- 组名允许带空格，但忽略前后空格
  例：某行“ [abc]def] # 测试”，则组名=“abc]def”
- 不含=的项名直接忽略即可（不读取，也不必报错）

【工具函数集的定义】

class cfgfile_read_tools 的定义放在 class_crt.h 中，各成员函数的说明如下：

★ cfgfile_read_tools()

使用说明：构造函数，共两个，分别适用于 char *和 string 形式的文件名

★ ~cfgfile_read_tools()

使用说明：析构函数，按需放入需要的内容

★ bool is_open()

使用说明：判断配置文件是否打开成功

★ int get_all_group(vector <string>& ret)

使用说明：返回配置文件中的所有组，放在 vector 中

- 成功返回组的数量，失败返回-1

★ int get_all_item(const char* const group_name,
vector <string>& ret, 放在 vector 中
const bool ignore_lower_upper_case = false)

使用说明：返回 group_name 组中的所有项,, 放在 vector 中

- ignore_lower_upper_case 为 true 表示 group_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回该组中项的数量，失败返回-1
- 还有一个重载函数，适合 string 类型的 group_name，其余同

★ int get_one_item(const char* const group_name,
const char* const item_name,
string &ret,
const bool ignore_lower_upper_case = false)

使用说明：返回 group_name 组中的 item_name 项的内容，放在 string 中

- ignore_lower_upper_case 为 true 表示 group_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回 0，失败返回-1
- 返回形式是“项名 = 值”的完整形式（供后续自行处理）
- 还有一个重载函数，适合 string 类型的 group_name/item_name，其余同

★ int item_get_value(const char* const group_name,
const char* const item_name,
const char* const def_str,
char* str,
const int str_maxlen,
const bool ignore_lower_upper_case = true)

使用说明：取 group_name 组中 item_name 的值，返回为 char *形式

- ignore_lower_upper_case 为 true 表示 group_name/item_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回 0，失败返回-1
- 返回值放在 str 中，限定最大长度（含尾零）为 str_maxlen（调用者要保证传入足够空间，不越界）
- 如果取不到，则返回 def_str（例：“name = ”形式）
- 如果值是含空格、tab 的字符串，则返回第一个空格、tab 前的内容即可

★ int item_get_value(const char* const group_name,
const char* const item_name,
const char* const def_str,
string& value,
const bool ignore_lower_upper_case = true)

使用说明：取 group_name 组中 item_name 的值，返回为 string 形式

- ignore_lower_upper_case 为 true 表示 group_name/item_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回 0，失败返回-1
- 返回值放在 value 中
- 如果取不到，则返回 def_str（例：“name = ”形式）
- 如果值是含空格、tab 的字符串，则返回全部内容

★ int item_get_value(const char* const group_name,
const char* const item_name,
char &ch,
const bool ignore_lower_upper_case = true)

使用说明：取 group_name 组中 item_name 的值，返回为 char 形式（单字符）

- ignore_lower_upper_case 为 true 表示 group_name/item_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回 0，失败返回-1
- 返回值放在 ch 中
- 如果取不到，则返回空格（例：“name = ”形式）

★ int item_get_value(const char* const group_name,
const char* const item_name,
const int min_value,
const int max_value,
const int def_value,
int& value,
const bool ignore_lower_upper_case = true)

使用说明：取 group_name 组中 item_name 的值，返回为 int 形式

- ignore_lower_upper_case 为 true 表示 group_name/item_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回 0，失败返回-1
- 返回值放在 value 中
- 如果不在[min_value..max_value]范围内/取不到/数据格式不正确，则返回 def_value

★ int item_get_value(const char* const group_name,
const char* const item_name,
const double min_value,
const double max_value,
const double def_value,
double& value,
const bool ignore_lower_upper_case = true)

使用说明：取 group_name 组中 item_name 的值，返回为 double 形式

- ignore_lower_upper_case 为 true 表示 group_name/item_name 大小写不敏感，false 表示大小写敏感，默认为 false
- 成功返回 0，失败返回-1

- 返回值放在 value 中
- 如果不在[min_value..max_value]范围内/取不到/数据格式不正确，则返回 def_value

【实现要求：】

1、在 BigHW 中新建项目 test_readcfg（注意：下划线）
   a) 附件中的 test_readcfg.cpp 放入 test_readcfg 中
   b) 附件中的 class_crt.h 放入 include 中
   c) 附件中的 class_crt.cpp 放入 common 中
   d) 通过 test_config_tools.cpp 的测试
2、参数分析用综合题 3
3、鼓励合理拆分源程序文件、合理划分函数、合理共用公共函数等
4、修改 common/include 中的内容后，要保证之前的 90-01-b*/90-02-b*能编译通过并运行正确
5、整个程序，不允许使用任何形式的全局变量/数组/指针，允许使用全局的宏定义或常变量
6、整个程序，不允许使用 goto
7、配置文件有 Windows/Linux 两种格式，均需要支持
8、提供 test_readcfg.exe 供参考
9、附件中的“参考配置- *.conf/*.ini”为配置文件参考样例，有 Windows/Linux 两种格式

【提交要求（仔细阅读，当心 0 分!!!）：】

1、提交作业前，先做好完整备份
2、之前大作业的lib记得删掉
3、要保证BigHW的所有项目都能编译通过
4、删除两个汉字库文件（作业检查时，会把字库文件放在exe所在目录下，要能保证打开）
5、按之前的BigHW提交要求，整个BigHW目录压缩成BigHW.rar，再按网页要求改名后提交

【编译器要求：】

仅 VS2022 通过即可

【作业要求：】

1、12 月 24 日前网上提交本次作业
2、每题所占平时成绩的具体分值见网页
3、超过截止时间提交作业则不得分

【Final：】

基于本学期多个工具集的综合应用