

【注意：】 1、程序中打开的文件，必须自行关闭

2、cpp 后缀的文件采用 C++方式(fstream)，c 后缀的采用 C 方式(FILE *)

补充：

7、文件 game.dat 为某游戏的存档文件，要求写一个能对存档文件中各项目进行修改的程序

【要求：】

(1) game.dat 的大小为 64 字节，具体的含义值及每个值的正确范围如下

64 个字节的解释如下：

- 1-16 : 玩家的昵称，最多 15 个字符
- 17-18: 生命值(0-10000)
- 19-20: 力量值(0-10000)
- 21-22: 体质(0-8192)
- 23-24: 灵巧(0-1024)
- 25-28: 金钱数量(0-100000000)
- 29-32: 名声值(0-1000000)
- 33-36: 魅力值(0-1000000)
- 37-44: 游戏累计时间(0-10000000000000000)
- 45 : 移动速度(0-100)
- 46 : 攻击速度(0-100)
- 47 : 攻击范围(0-100)
- 48 : 预留值，暂不用
- 49-50: 攻击力(0-2000)
- 51-52: 防御力(0-2000)
- 53 : 敏捷度(0-100)
- 54 : 智力(0-100)
- 55 : 经验(0-100)
- 56 : 等级(0-100)
- 57-58: 魔法值(0-10000)
- 59 : 使用魔法时每次消耗的魔法值(0-100)
- 60 : 魔法伤害力(0-100)
- 61 : 命中率(0-100)
- 62 : 魔法防御力(0-100)
- 63 : 暴击率(0-100)
- 64 : 耐力(0-100)

注：本文档中的字节编号从 1 开始，取值均为整型（1/2/4/8 字节长度）

- (2) 假设 game.dat 放在当前目录下，即不需要带路径
- (3) 要求所有值都能修改，但修改只能在每个值的合理范围内
- (4) 程序运行时，支持两个参数，参数--read 可以读取 game.dat 的内容并在屏幕上显示，用于验证修改是否正确，参数--modify 可以进入修改模式
- (5) 给出一个示例 game.dat 供参考(16 进制查看与编辑可用 UltraEdit 软件)
- (6) 提供一个 15-b7-demo.exe 供参考
- (7) 要求--read 的输出与 demo 完全一致
- (8) 修改模式(--modify)下运行时先从 game.dat 中读取全部 64 字节数据，再用简易菜单方式列出各项（每项给出当前值），用户可以选某项后进行具体修改；可以多次选择多次修改，包括对同一项多次修改；退出有两个选项：放弃修改/存盘退出，如果选择存盘退出，则更新 game.dat 文件，否则不更新

(9) 本题要求两种形式实现:

★ 采用特定的 C++ 文件读写方法, 包括:

- 程序 **只准** 在最开始 open 一次, 最后 close 一次 (不含出错后 close), 中间 **不准** close 后再 open
- 读写数据只准用 read/write 函数, 其余 (按字符读写、按行读写、格式化读写等, 包括 peek) 均违规 (good/fail/seek/tell/eof 等非读写的辅助操作不限)
- 提供基准程序 15-b7-1.cpp, 不要违反限制规定

★ 采用特定的 C 文件读写方法, 包括:

- 程序只准在最开始 fopen 一次, 最后 fclose 一次 (不含出错后的 fclose), 中间不准 fclose 后再 fopen
- 读写数据只准用 fread/fwrite 函数, 其余 (按字符读写、按行读写、格式化读写等, 包括 peek) 均违规 (ftell/fseek、feof 等非读写的辅助操作不限)
- 提供基准程序 15-b7-2.c, 不要违反限制规定

★ **违反读写要求的, 作业总分-10**

8、bmp 图像格式读取及图形显示

【题目背景:】

bitmap 是一种图像格式, 后缀为.bmp, 是 Windows 操作系统下的标准图像文件格式; 它以非压缩方式进行数据的存储 (不必关注压缩算法), 基本结构为文件头信息+图像头信息+颜色表 (可选)+各点阵的颜色信息。

【题目要求:】

- 自行查阅相关资料 (用类似 bmp 文件结构、格式分析、格式解析等关键字搜索), 看懂 bmp 文件的基本存储格式, 然后读出各点阵信息, 用 hdc_point 将图像在 cmd 窗口下还原出来
- 要求支持的格式为 bmp 的单色、16 色、256 色、24 位色、32 位色五种
- 需要自行调整 cmd 窗口的大小来保证图片显示完整 (作业检查时图片控制在 1366x768 以内)
- **不需要** 考虑缩放
- **不考虑** 显示窗口最大化/最小化/移动带来的图像缺失
- 通过命令行带参数方式实现, 具体参数要求为:
 - help : 查看帮助信息
 - bmpfile ****.bmp : 指定 bmp 图片文件名, 要考虑路径, 不考虑输入错误
 - start_x 0-300 : 指定图片左上角显示时的 x 坐标 (cmd 窗口的左上角为 0)
 - start_y 0-300 : 指定图片左上角显示时的 y 坐标 (cmd 窗口的左上角为 0)
 - angle 0/90/180/270 : 指定图片旋转角度, 方向为逆时针, 默认 0
 - mirror : 指定图片镜像显示 (以 0° 旋转角为基准, 左右互换)
- 附件给出了三个头文件/源程序文件
 - 15-b8-main.cpp: 给出了画点的基本方法、参数分析及 bitmap 的初始化, 不允许改动, **不需要提交**
 - 15-b8-bmp.h : 给出一个 bitmap 类, 按要求补充完成的即可 (**需提交**)
 - 15-b8-bmp.cpp : 具体的格式分析、读取像素信息的实现 (**需提交**)
- 给出 15-b8-demo.exe 供参考
- 显示顺序不限制 (demo 为从上到下, 从左到右)

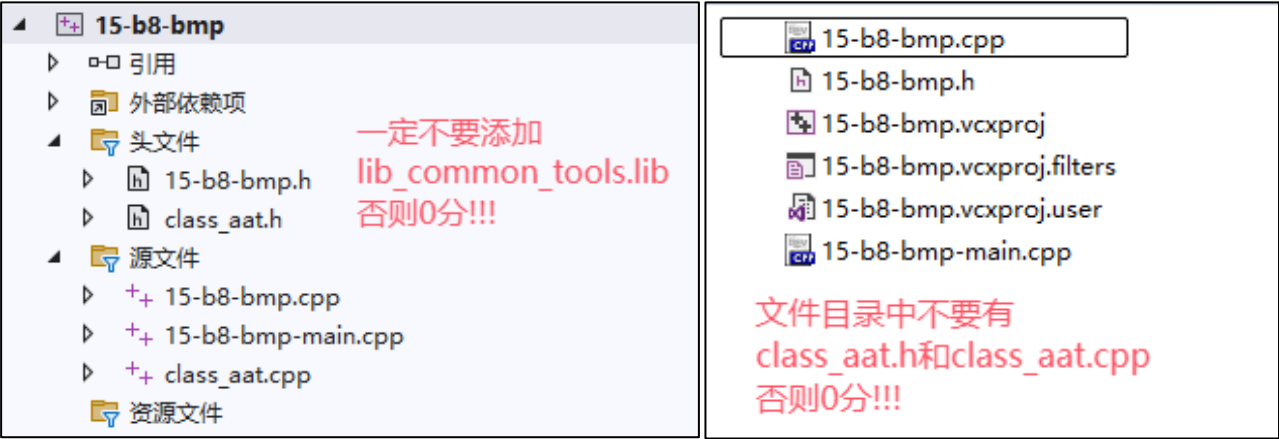
【特别限制:】

- bitmap 是 Windows 的标准图像文件格式, 在 <Windows.h> 中有很好的支持, 但是为了更深入的掌握二进制文件的读写, 在 15-b8-bmp.h 及 15-b8-bmp.cpp 两个文件中, **不允许 #include <Windows.h>, 否则期末总分-20!!!**
- 用什么方式定义存储结构可以快速读取文件头信息, 而不是一个个字节去拼凑? 具体可查看 15-b8-bmp.h 中的注释, 看能否得到某些启发

【目录要求:】

因为要用到 main 函数的参数分析工具，因此本作业放到大作业中完成，但分数计入小作业，具体要求如下：

- 在 BigHW 下建立一个 15-b8-bmp 的项目（都是中减号，不是下划线!!!）
- 将附件提供的 15-b8-bmp.h、15-b8-bmp.cpp、15-b8-bmp-main.cpp 放入项目中
- 再将第三次大作业的 class_aat.h 和 class_aat.cpp 添加到项目中（**注意：**文件不要放入项目中，否则 0 分!!!）
- class_aat.h 中的 ENABLE_LIB_COMMON_TOOLS 确保为 0
- 项目结构及项目目录如下图所示



【提交要求:】

本次作业需要在**两个位置进行提交**

- 第 15 章中提交 15-b8-bmp.h、15-b8-bmp.cpp 两个文件，作业检查时会用老师的 class_aat 工具集参与编译，目的是检查你的 bmp 图像分析是否正确
- 大作业中再次提交整个 BigHW 的压缩包（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），目的是检查你的参数分析是否完成

【编译器要求:】

		编译器VS	编译器Dev	编译器Linux
15-b7-1.cpp	游戏配置文件读写（C++）	Y	Y	Y
15-b7-2.c	游戏配置文件读写（C）	Y	Y	Y
15-b8-bmp.h	bmp图像读取-头文件	Y	/	/
15-b8-bmp.cpp	bmp图像读取-源文件	Y	/	/

【作业要求:】

- 1、**11 月 21 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会自动扣除相应的分数，具体见网页上的说明 •