

- 【注意：】** 1、程序中打开的文件，必须自行关闭
 2、无明确要求，则文件读写采用 C++ 方式，不允许采用 C 方式

补充：

2、在完成基础知识题的基础上，写一个测试程序，能测试当前的文件是 Windows 格式还是 Linux 格式，也能将某种格式转换为另一种格式

【要求：】

(1) 通过 main 函数带参数的方式分别实现不同功能，具体如下

- 17-b2 --check a.txt : 检查 a.txt 的格式，输出为“Windows 格式” / “Linux 格式” / “文件格式无法识别” / “文件无法打开”四种
- 17-b2 --convert wtol a.txt b.txt : 将 Windows 格式的文件 a.txt 转换为 Linux 格式的文件 b.txt (如果 a.txt 不是 Windows 格式，给出“文件格式无法识别”信息即可)
- 17-b2 --convert ltow c.txt d.txt : 将 Linux 格式的文件 c.txt 转换为 Windows 格式的文件 d.txt (如果 c.txt 不是 Linux 格式，给出“文件格式无法识别”信息即可)

(2) 判断标准：

- 如果所有的 0x0A 前均有 0x0D，则认为是 Windows 格式 (多个 0x0D 后跟一个 0x0A，算)
- 如果所有的 0x0A 前均无 0x0D，则认为是 Linux 格式

(3) 给出 17-b2-demo.exe 供参考

3、写一个文件的 16 进制查看器，要求输入一个文件，按格式打印它的每个字节的值

【要求：】

(1) 文件允许任何类型，既可以是文本文件，也可以是二进制文件

(2) 通过 main 函数带参数的方式分别实现不同功能，具体如下

- 17-b3 --infile a.txt : 将 a.txt 文件按 hex 格式输出到屏幕上
 - 17-b3 --infile a.txt --outfile b.hex : 将 a.txt 文件按 hex 格式转换到 b.hex 中
 - 17-b3 --outfile b.hex --infile a.txt : 同上 (即 --outfile/--infile 顺序不限)
- (3) 输出时，每行显示 16 个原始字节，分三部分，具体格式如下
- 左侧是该行首字符在文件中的位置 (16 进制，宽度 8 位)
 - 间隔 2 个空格
 - 中间是每个字节的十六进制形式，每个字节间空格分割，每 8 个中间有“-”
 - 间隔 5 个空格
 - 右侧为每个字节的 ASCII 字符形式 (ASCII 码在 33-126 之间图形字符打印，其余用. 表示)

建议：程序完成后，可以看看将. 替换为实际字符，会有什么样的显示效果

(4) 提供 17-b3-demo.exe 供参考

(5) 显示完成后，字节数与实际文件字节数不能有差异

(6) 可以用 UltraEdit 或 Notepad++ 等编辑软件，打开输入文件，选择 16 进制方式查看文件，然后与你的程序进行比对，来确定正确性

【提示：】

- (1) 打开文件时，ios::binary 选项加不加，是否有区别？
- (2) 二进制文件是否结束的判断和文本文件是否有区别？

4、写一个 17-b3 的逆转换程序，将被转换为 16 进制查看格式的文件重新转换为原格式

【要求：】

(1) 通过 main 函数带参数的方式分别实现不同功能，具体如下

- 17-b4 --infile b.hex --outfile a.bin : 将 hex 格式的 b.hex 转换为 a.bin
- 17-b4 --outfile a.bin --infile b.hex : 同上（即--outfile/--infile 顺序不限）
- 本题的--infile 就是 17-b3 的--outfile，格式要求相同，不考虑错误格式
- 本题的--outfile 和 17-b3 的--infile 应该完全一致，即在 Windows 的 cmd 下运行 `comp a.txt a.bin` / 在 Linux 下运行 `diff -s a.txt a.bin`，应该报完全相同）

(2) 提供 17-b4-demo.exe 共参考

(3) 17-b3 与 17-b4 要求能够同学间双向验证，每人的 17-b4 需要验证至少 3 人的 17-b3，将验证名单放在 17-b4 源程序的第 2 行用注释说明即可（正常情况双向查验都应该正确，如果查验不正确则要连环扣分，即生成者还原者同步扣分）

5、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(0-100000000000000000)
- 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 17-b5-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) 本题要求三种形式实现：
 - 9.1 自定义方式，要求用 C++的文件处理，但具体读写方式等不限
 - 9.2 采用特定的 C++文件读写方法，包括：
 - 9.2.1 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
 - 9.2.2 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/seek/tell/eof 等非读写的辅助操作不限）
 - 9.2.3 提供基准程序 17-b5-2.cpp，不要违反限制规定
 - 9.3 采用特定的 C 文件读写方法，包括：
 - 9.2.1 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
 - 9.2.2 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）
 - 9.2.3 提供基准程序 17-b5-3.c，不要违反限制规定
 - 9.4 **违反读写要求的，作业总分-10**

【编译器要求:】

		编译器VS	编译器Dev	编译器Linux
17-b2.cpp	Windows/Linux文件格式测试及互转	Y	Y	Y
17-b3.cpp	文件的16进制查看器	Y	Y	Y
17-b4.cpp	16进制查看器逆转换	Y	Y	Y
17-b5-1.cpp	改游戏配置文件（C++方式，任意）	Y	Y	Y
17-b5-2.cpp	改游戏配置文件（C++方式，限定）	Y	Y	Y
17-b5-3.cpp	改游戏配置文件（C方式，限定）	Y	Y	Y

【作业要求:】

- 1、**5月29日（周六）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