目标：

**构件库部分：**

为双向链表类添加如下功能:

1. 按武力从大到小排序的功能。
2. 按综合值(各项指标之和)从大到小排序的功能。

**应用验证部分：**

1. 从文本文件“三国人物.txt”中读取人物信息，并存储在双向链表中。
2. 输出按武力从大到小排序的三国人物表。
3. 输出按综合值从大到小排序的三国人物表。

**提高要求：**

将双向链表类改为双向链表模板，使得基础数据的类型可以是任意的，并进行应用验证。

原理：

编写构件库程序，定义一个双向链表类，具体如下所示：

class FH\_Chain

{

private:

string m\_name;

int m\_strength;

int m\_intelligence;

int m\_power;

int m\_charisma;

int m\_luck;

FH\_Chain \* m\_previous; //空链表m\_previous==NULL

FH\_Chain \* m\_next;

public:

FH\_Chain();

~FH\_Chain();

FH\_Chain & operator [ ] (int i);

void CreateNode();

void DeleteNode(FH\_Chain \* p);

int GetNodeNumber();

void AllPutOut();

void SwapWith(FH\_Chain & destination, FH\_Chain & source);

void SortAll(SortType s);

friend void CopySingleNode(FH\_Chain & destination, FH\_Chain & source);

friend void CreateNodeFromFile(FH\_Chain & r, string fileName);

friend ostream & operator << (ostream & stream, FH\_Chain & myChain);

};

该双向链表类相关的两个重要函数如下所示：

extern void CopySingleNode(FH\_Chain & destination, FH\_Chain & source);

extern void CreateNodeFromFile(FH\_Chain & r, string fileName);

为了排序函数方便判断排序标准，定义：

enum SortType { PowerSort, AllSort, SortTypeNum };

其中实现了重载运算符 [ ] 与输出流运算符 << ，实现了交换两个节点的函数与排序函数，其余函数参考了雍老师课件（第13讲：双向链表）。在排序函数中，使用了冒泡排序法，同时调用交换函数。同时在需要的时候使用了try…catch语句，避免程序崩溃。

编写手动测试与自动测试函数，分别提供不同的运行接口。在自动测试部分，使用输出重定向，将输出到显示器的部分重定向到文本文件中，方便用文件输入流进行比较判断程序运行错误与否。

**如何编程和运行：**

* 软件构件库部分：

编写原理中所述函数的具体实现。

* 构件库测试部分：

编写原理中所述自动测试与手动测试部分的代码实现。

* 构件库应用部分：

建立工程，编写 main() 函数并且选择手动测试与自动测试的功能，之后调用类中的函数进行操作。

结果：

以下测试结果均在64位操作系统下用VS 2017 测试得到。

程序验证结果报告：

* 对于64位的计算机

在上述的程序运行过程中未发现错误，在程序运行过程中可以方便地发现程序的错误报告，同时自动测试的部分显示重点突出，方便使用。

结论：

程序能较好完成设计目的要求，在使用的过程中未发现错误。

模板类部分：

将上述程序使用模板类的形式进行封装，实现了一个比较通用的双向链表类，其中具体数据引入另一个类（Data），需要对Data进行输入流输出流运算符和比较运算符的重载。

模板类中实现了排序、复制结点、交换结点、输出链表以及一些比较基础的构造、析构等几个功能，在这个基础上我们可以比较方便的使用。

在模板类的验证过程中，也使用了同样的三国人物文件，排序时只采用综合排序方法（可以在Data类的比较运算符重载中修改）。同时有自动测试与手动测试功能。

需要注意，**模板类的函数实现与正常类不同**，直接类外实现会有问题。

C 与 C++ 双向链表实现的不同：

1. C语言中需要使用结构体，没有析构函数，需要手动保证每一次都**释放内存**；
2. C语言中不能使用模板类，需要每一次单独编写双向链表；
3. C语言中需要使用指针，不如引用方便。