目标：

编写程序，从文本文件“data.txt”读取一系列整数（文件格式请自行定义），并采用向量vector进行存储。然后，分别实现:

1. 采用算法(algorithm)库的sort函数进行排序(不去重)，并输出排序结果。
2. 采用集合(set)进行排序(去重)，并输出排序结果。

原理：

编写构件库程序，定义一个排序类，具体如下所示：

class FH\_Sort\_Class

{

private:

vector<long long> m\_Vec;

set<long long> m\_Set;

void Read();

void Print();

public:

FH\_Sort\_Class();

~FH\_Sort\_Class();

void Start();

};

其中在Read( ) 函数中读入data.txt的信息，存储在m\_Vec和m\_Set中，而在Print( ) 函数中实现排序后数据的输出功能。而Start( ) 函数对上述两个函数进行了封装并且提供public类型的接口，方便在外部进行统一调用。

同时声明了错误类型

enum ErrorType { NoSuchFile, NoThing, ErrorReading };

在类中的函数里面使用try catch语句，并且分情况输出错误信息，避免程序崩溃。同时，在NoSuchFile错误出现时会提供现在需要放置data.txt的工作目录（根据debug模式与否区分，分别给出不同的目录），方便调试。

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：

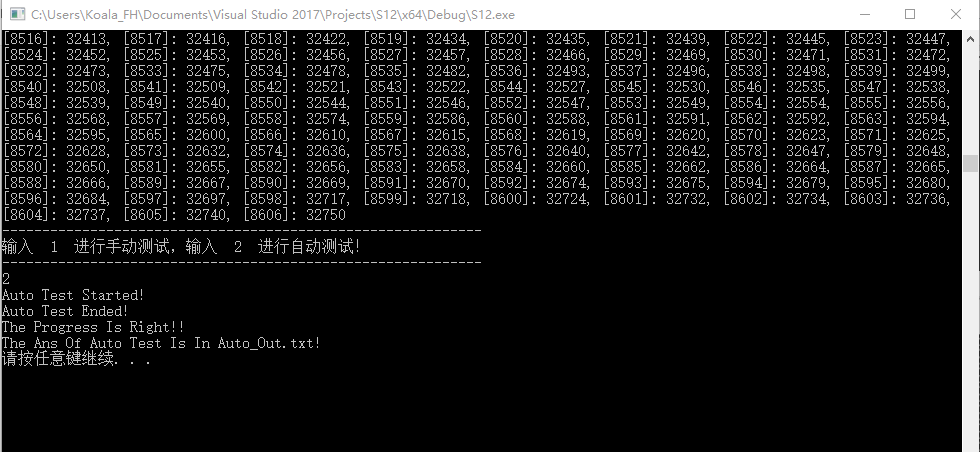
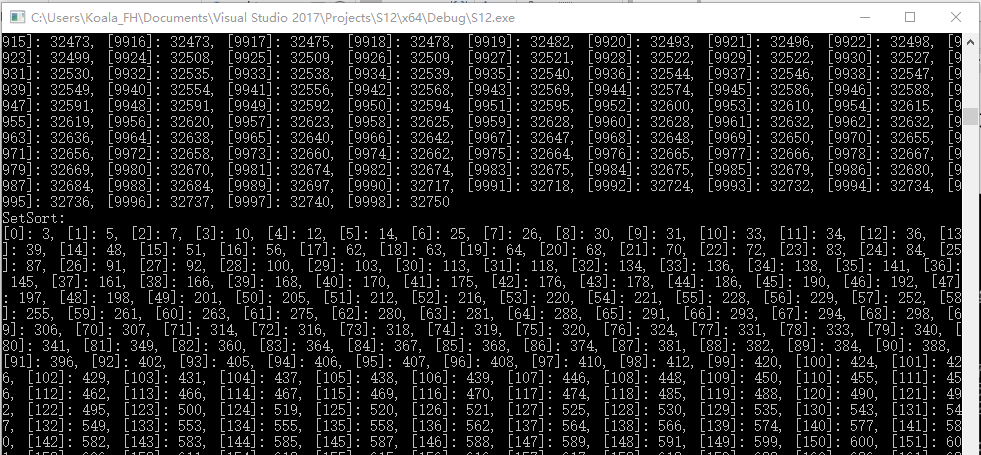
编写原理中所述自动测试与手动测试部分的代码实现。

* 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。

由于程序处理数据较多，产生的记录较长，只展示一部分运行结果截图。

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。