目标：

编写C++程序来达到：编写程序，可以接受从控制台窗口输入的源程序代码文件名。对于在该文件中的每个分号，如果该分号之后不是回车或换行符，则自动在该分号之后添加回车和换行符。统计出现这种情况的分号个数，并在控制台窗口中输出统计结果，同时将统计结果保存到日志文件文件“log.txt”中。

原理：

编写构件库程序，在其中放置可以用于检测并且优化代码分号换行的函数（该函数可以指定输出目录，可以接受文件名与完整目录文件名（不需要将\换为\\）。）与改写路径名的函数（将\自动变为\\，同时防止多个\在一起造成的影响）。在这里考虑了多个分号连续的情况与分号后紧跟着回车的情况，并且尽量不使新生成的文件将原有文件替代掉。

同时，在输出记录文件log.txt时，同时会记录运行时间、源文件路径、生成文件路径以及查找出的错误数目。

在手动测试部分，如果所查找的文件不存在，会有提示并且要求重新输入。

在自动测试部分，准备了样例，在运行结束后将运行结果和参考答案进行比对（包括找出的错误数目以及修改后的程序代码）。

**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：

编写所需要的测试函数，并且在其中内置样例。

* 构件库应用部分：

编写main函数，在函数中提供选择自动和手动测试的功能。

结果：

以下测试数据均在64位操作系统下用VS 2017 测试得到。

自动测试程序运行结果：

------------------------------------------------------------

输入 1 进行手动测试，输入 2 进行自动测试！

------------------------------------------------------------

2

------------------------------------------------------------

自动测试开始！

------------------------------------------------------------

第 1 组运行正确！发现14处错误！

第 2 组运行正确！发现45处错误！

第 3 组运行正确！发现13处错误！

第 4 组运行正确！发现93处错误！

第 5 组运行正确！发现34处错误！

第 6 组运行正确！发现45处错误！

------------------------------------------------------------

请按任意键继续. . .

手动测试程序某一次运行结果：

------------------------------------------------------------

输入 1 进行手动测试，输入 2 进行自动测试！

------------------------------------------------------------

1

------------------------------------------------------------

请输入一个文件名（与exe文件在一起），或者输入文件的完整路径（路径中的\不需修改为\\）！

请注意，如果在VS下直接运行，请将文件放在S07\S07\目录下；如果直接运行exe文件，请将要检查的文件放在exe文件相同的目录下！

a.txt

文件目录：C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\Debug\\a.txt

操作成功！输出到文件\_Changed\_a.txt ，其中找到错误49个！

------------------------------------------------------------

------------------------------------------------------------

输入 1 进行手动测试，输入 2 进行自动测试！

------------------------------------------------------------

1

------------------------------------------------------------

请输入一个文件名（与exe文件在一起），或者输入文件的完整路径（路径中的\不需修改为\\）！

请注意，如果在VS下直接运行，请将文件放在S07\S07\目录下；如果直接运行exe文件，请将要检查的文件放在exe文件相同的目录下！

b.txt

文件目录：C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\Debug\\b.txt

操作失败！找不到文件！请重新输入！

------------------------------------------------------------

请输入一个文件名（与exe文件在一起），或者输入文件的完整路径（路径中的\不需修改为\\）！

请注意，如果在VS下直接运行，请将文件放在S07\S07\目录下；如果直接运行exe文件，请将要检查的文件放在exe文件相同的目录下！

g:\a.txt

文件目录：g:\\a.txt

操作成功！输出到文件\_FH\_Changed\_a.txt ，其中找到错误45个！

------------------------------------------------------------

经过上述测试过程后的log.txt文件：

于2017年4月15日15时14分16秒运行程序，检查文件 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\1.cpp ，最终输出为 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\\_Changed\_1.cpp ，其中找到错误共 14条！

于2017年4月15日15时14分16秒运行程序，检查文件 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\2.cpp ，最终输出为 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\\_Changed\_2.cpp ，其中找到错误共 45条！

于2017年4月15日15时14分16秒运行程序，检查文件 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\3.cpp ，最终输出为 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\\_Changed\_3.cpp ，其中找到错误共 13条！

于2017年4月15日15时14分16秒运行程序，检查文件 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\4.cpp ，最终输出为 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\\_Changed\_4.cpp ，其中找到错误共 93条！

于2017年4月15日15时14分16秒运行程序，检查文件 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\5.cpp ，最终输出为 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\\_Changed\_5.cpp ，其中找到错误共 34条！

于2017年4月15日15时14分16秒运行程序，检查文件 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\6.cpp ，最终输出为 C:\\Users\\Koala\_FH\\Documents\\Visual Studio 2017\\Projects\\S07\\x64\\TestData\\\_Changed\_6.cpp ，其中找到错误共 45条！

于2017年4月15日15时19分40秒运行程序，检查文件 a.txt ，最终输出为 \_Changed\_a.txt ，其中找到错误共 49条！

于2017年4月15日15时20分9秒运行程序，检查文件 g:\\a.txt ，最终输出为 \_FH\_Changed\_a.txt ，其中找到错误共 45条！

其中一个未修改的文件：

#include "FH\_Find.h"

string Changed;int GetEndLine(string FilePath)

{

ifstream FH\_File\_In(FilePath, ios::in);if (!FH\_File\_In)

return -1;string TargetFile = "\_Changed\_" + Get\_File\_Name(FilePath);ofstream FH\_File\_Out;int LoopTimes = 0;do

{

ifstream \*In\_Temp = NULL;In\_Temp = new ifstream(TargetFile, ios::in);if (!\*In\_Temp)

{

In\_Temp->close();delete In\_Temp;break;}

In\_Temp->close();delete In\_Temp;LoopTimes++;TargetFile = "\_FH" + TargetFile;} while (LoopTimes < 10);if (LoopTimes >= 10)

{

TargetFile = "\_Changed\_" + Get\_File\_Name(FilePath);}

Changed = TargetFile;FH\_File\_Out.open(TargetFile, ios::out);int No = 0;char ch = '0';while (ch != EOF)

{

ch = FH\_File\_In.get();FH\_File\_Out << ch;if (ch == ';')

{

FH\_Mark: ch = FH\_File\_In.get();if (ch != '\n' && ch != ';')

{

No++;FH\_File\_Out << '\n' << ch;}

else if (ch == ';')

{

No++;FH\_File\_Out << '\n' << ch;goto FH\_Mark;}

}

}

FH\_File\_In.close();FH\_File\_Out.close();return No;}

string Get\_File\_Name(string FilePath)

{

int i;string Ans = "";for (i = FilePath.length() - 1; i >= 0; --i)

if (FilePath[i] == '\\')

break;i++;for (; i < FilePath.length(); ++i)

Ans += FilePath[i];return Ans;}

经过上述修改过程后的该文件：

#include "FH\_Find.h"

string Changed;

int GetEndLine(string FilePath)

{

ifstream FH\_File\_In(FilePath, ios::in);

if (!FH\_File\_In)

return -1;

string TargetFile = "\_Changed\_" + Get\_File\_Name(FilePath);

ofstream FH\_File\_Out;

int LoopTimes = 0;

do

{

ifstream \*In\_Temp = NULL;

In\_Temp = new ifstream(TargetFile, ios::in);

if (!\*In\_Temp)

{

In\_Temp->close();

delete In\_Temp;

break;

}

In\_Temp->close();

delete In\_Temp;

LoopTimes++;

TargetFile = "\_FH" + TargetFile;

} while (LoopTimes < 10);

if (LoopTimes >= 10)

{

TargetFile = "\_Changed\_" + Get\_File\_Name(FilePath);

}

Changed = TargetFile;

FH\_File\_Out.open(TargetFile, ios::out);

int No = 0;

char ch = '0';

while (ch != EOF)

{

ch = FH\_File\_In.get();

FH\_File\_Out << ch;

if (ch == ';

')

{

FH\_Mark: ch = FH\_File\_In.get();

if (ch != '\n' && ch != ';

')

{

No++;

FH\_File\_Out << '\n' << ch;

}

else if (ch == ';

')

{

No++;

FH\_File\_Out << '\n' << ch;

goto FH\_Mark;

}

}

}

FH\_File\_In.close();

FH\_File\_Out.close();

return No;

}

string Get\_File\_Name(string FilePath)

{

int i;

string Ans = "";

for (i = FilePath.length() - 1;

i >= 0;

--i)

if (FilePath[i] == '\\')

break;

i++;

for (;

i < FilePath.length();

++i)

Ans += FilePath[i];

return Ans;

}



程序验证结果报告：

* 对于64位的计算机

在上述的计算过程中未发现错误。

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。