MODE-GL实现细节

1. 问题描述

N个asset，选择其中K个组成一组投资组合，K个asset中，每个asset获得一个投资比例后都能计算出它的预估收益以及预估风险，K个asset的总投资比例为1，该投资组合问题要求最小化风险，最大化收益。

在该问题中，存在6种约束：

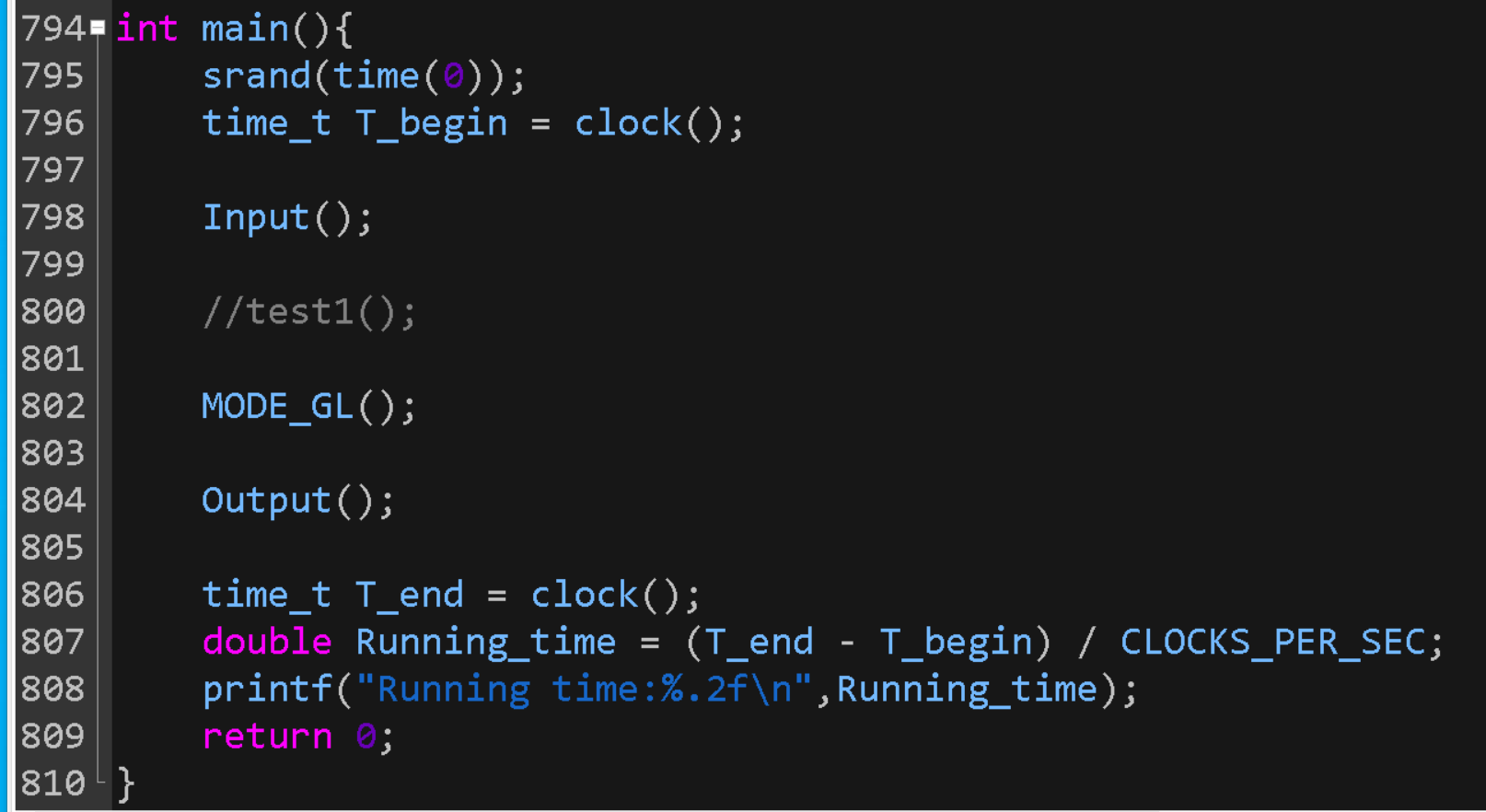
1. 基数约束：每种组合最多选择K个asset。
2. 上下限约束：在被选择的K个asset中，每一个asset的投资比例有一个下限与上限。
3. 预分配约束：有|Z|个asset已经被要求一定要选择。
4. 批约束：投资组合中的任何一个asset的投资比例都必须是vi的精确倍数，该vi是市场上能够交易的最小批数（或者用最小交易单元来理解）。
5. 类约束：N个asset，这些asset属于不同的类型（有些相同，有些不同），假设这N个asset一共有M种类型，则要求这M种类型中，每一种都满足至少一个属于该类的asset被选择在K中。
6. 类限约束：每一种类的总投资比例有上下限。
7. 输入输出描述

输入为94个asset的1000天实际收益记录，该组文件在当前目录下的MODED-GL-DataSets文件夹中。

输出为用A集合，该集合在下面的内容中会有描述。

1. 代码细节

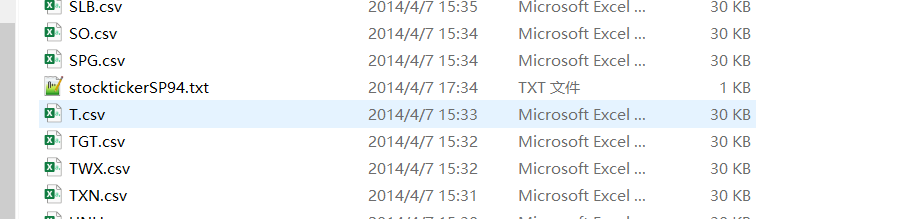
main函数：



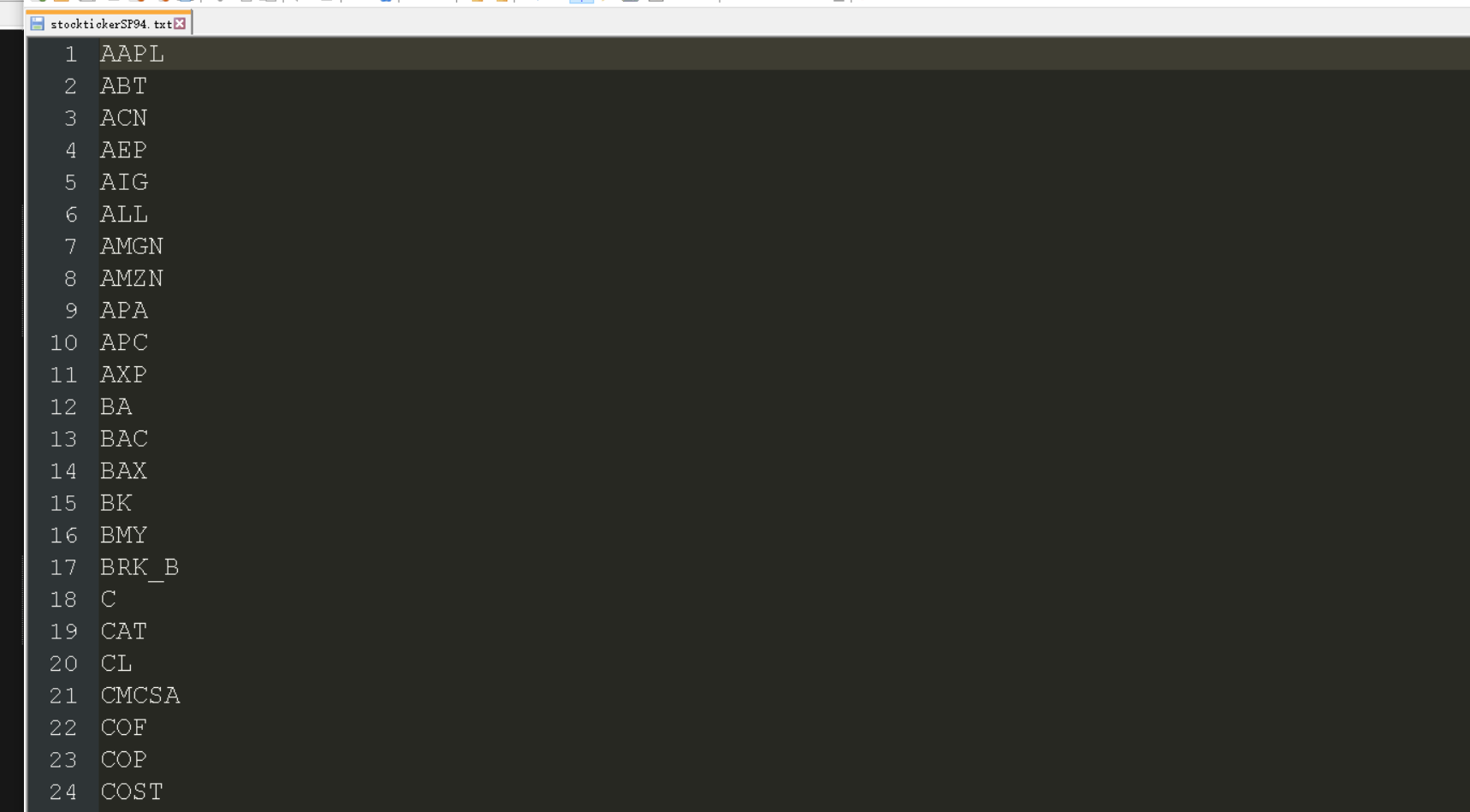
main函数主要由3个函数组成，分别是输入函数，输出函数以及处理数据的函数

srand(time(0))语句初始化随机数种子，其它内容用于计算程序运行时间

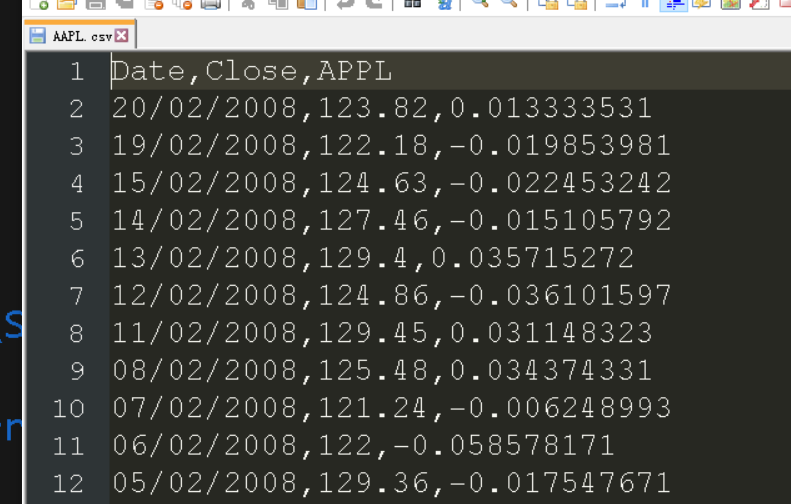
Input()函数用于输入当前目录下MODED-GL-DataSets/S&P/中S&P100文件，这些文件由数据文件以及数据文件名称文件。如下图：



.csv后缀的文件是数据文件，其中stocktickerSP94.txt文件是这些数据文件的名称。



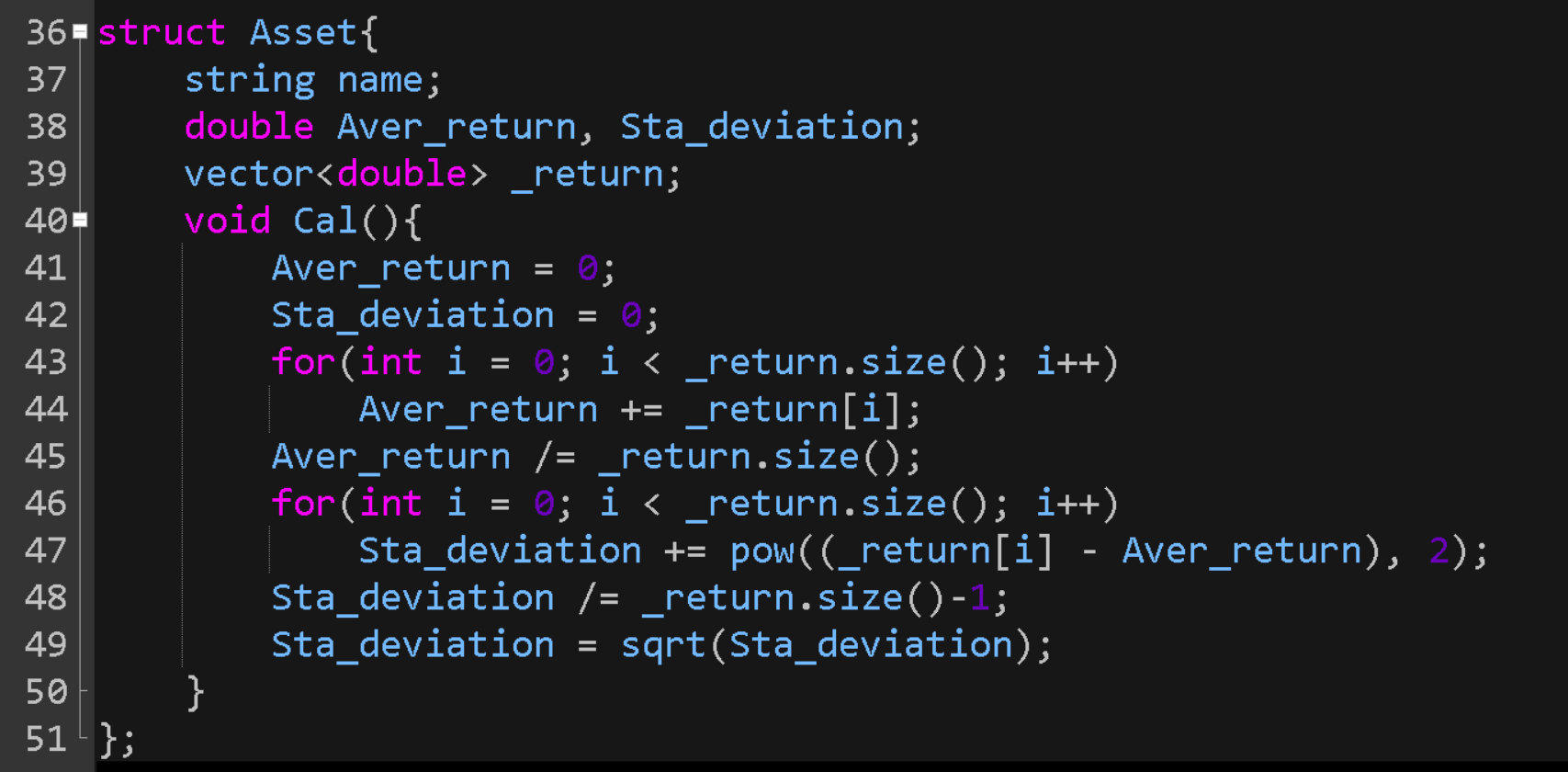
.csv数据文件的内容如下：



除了第一行，其他每行存在两个“,”将数据分为三列，第一列是时间，第二列是当天asset的收盘价，第三列是增长率（即（今天收盘价-昨天收盘价）/ 昨天收盘价）。我们只需要第二列的

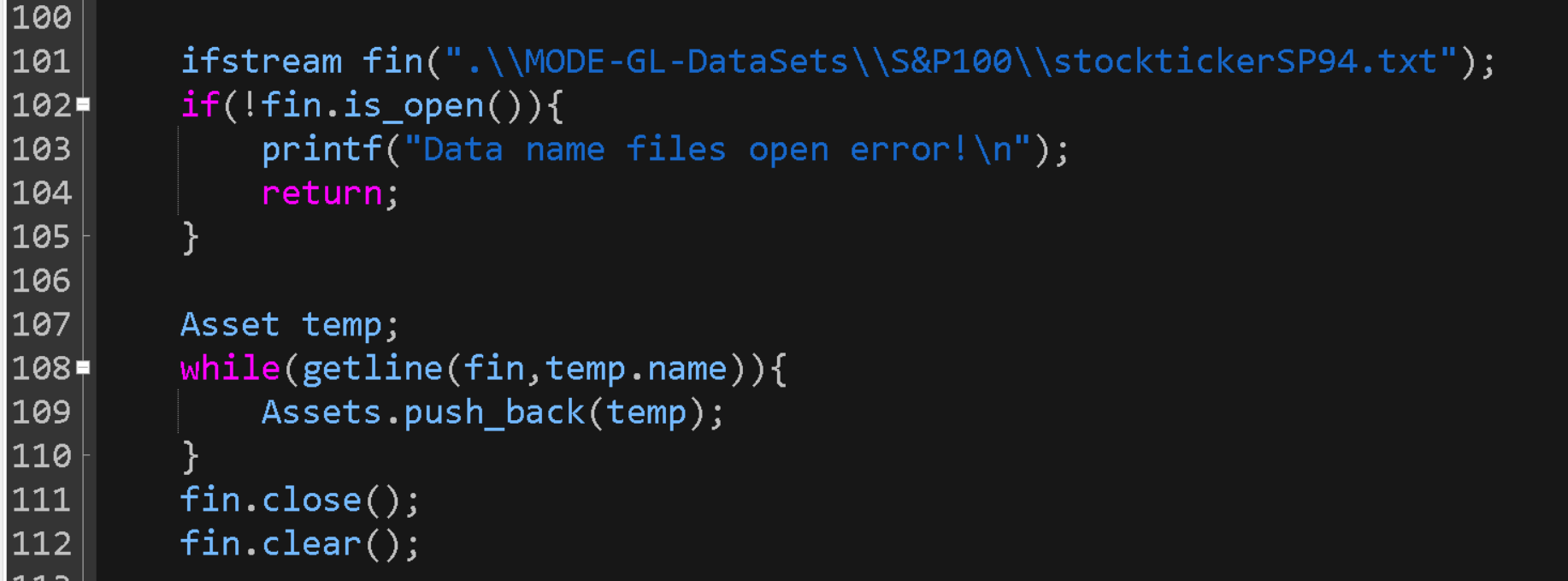
由这些数据，对输入函数的编写思路是：

1. 建立asset的结构体，该结构体里的属性有asset的名称，每日相对于前一天的收益数组，平均收益以及收益标准差。



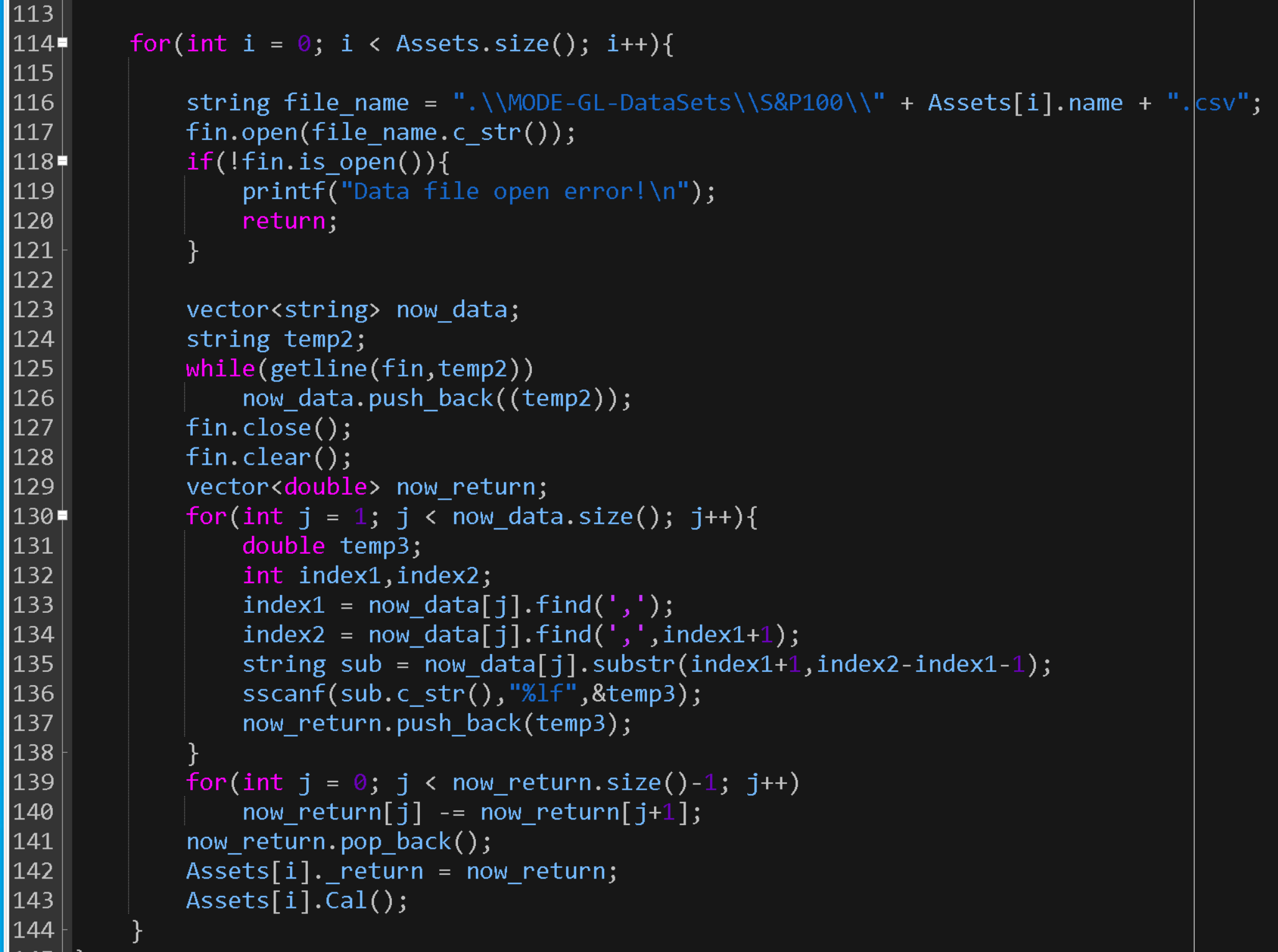
Cal()函数是该结构体的内置函数，用来计算一个asset的平均收益与标准差。

1. 通过读入stocktickerSP94.txt获得每个asset的名称。



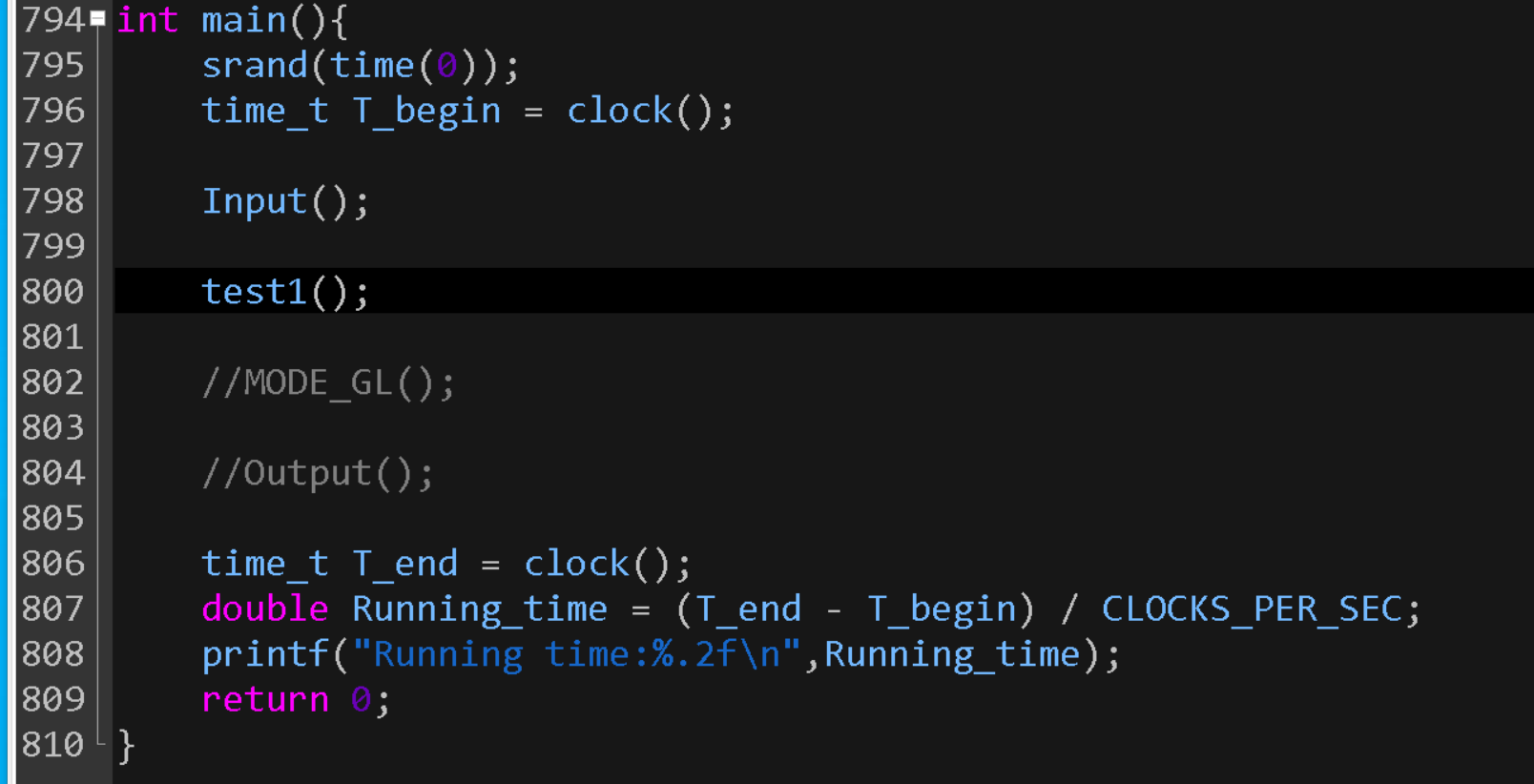
101-105打开数据名称文件，107-112读取该文件每行内容并且存入暂存入Asset类型变量中。109行出现的Assets是一个Asset类型的全局动态数组。

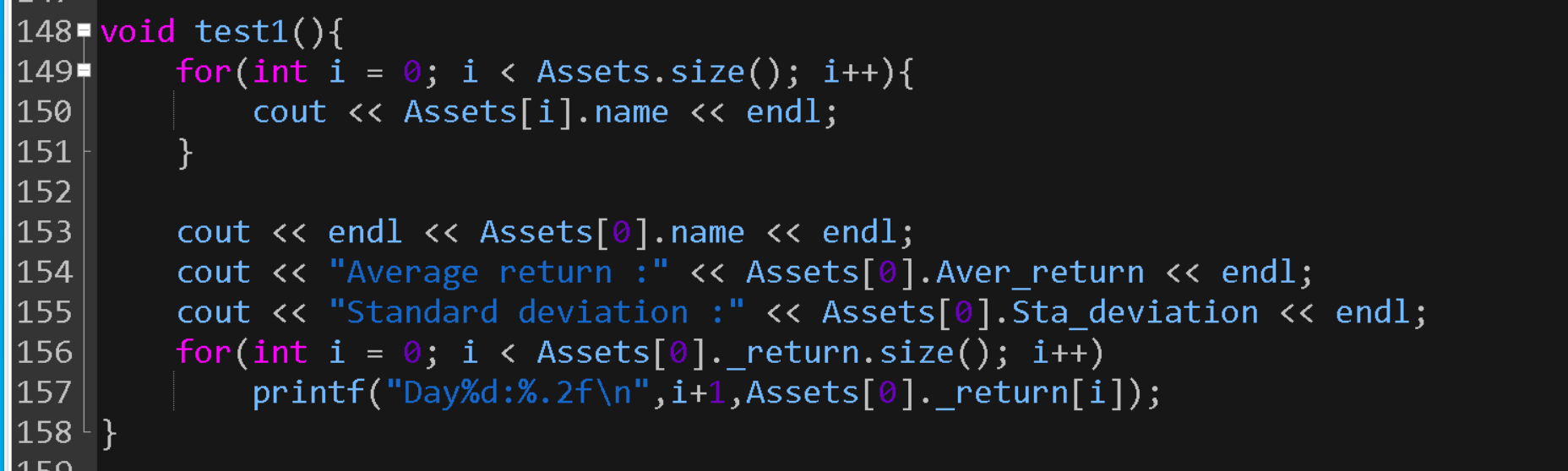
1. 通过上一步读入的数据文件名称打开每个asset数据文件，对于每个asset文件，使用一个string类型的动态数组存储该数据文件全部数据，然后对于第二行开始的数据，提取第二列的收盘价，存入当前asset结构体中的\_return数组，然后计算每一天的收益（今天收盘价-昨日收盘价）。因为日期是从后往前读取（即第j行的时间更晚，j+1行的时间更早），所以用\_return[j] - \_return[j+1]计算每一天的收益，计算完成之后，删除\_reutrn数组中的最后一个元素，并调用Asset结构体中的Cal()函数计算平均收益以及其标准差。



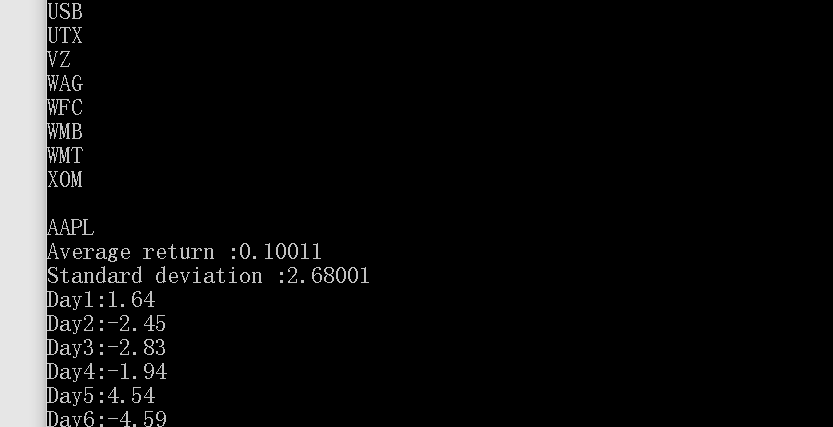
116-121打开文件，123-128读取内容，129-138提取从第二行开始的第二列数据，存入now\_return数组中，139-140计算每天收益以及删除最后一个元素，142行将当前提取的数据保存在Assets数组中，143调用Cal()函数。

现在回到主函数，测试是否能够正确读取数据，这里做了一个test1()函数用作测试数据是否读入成功。

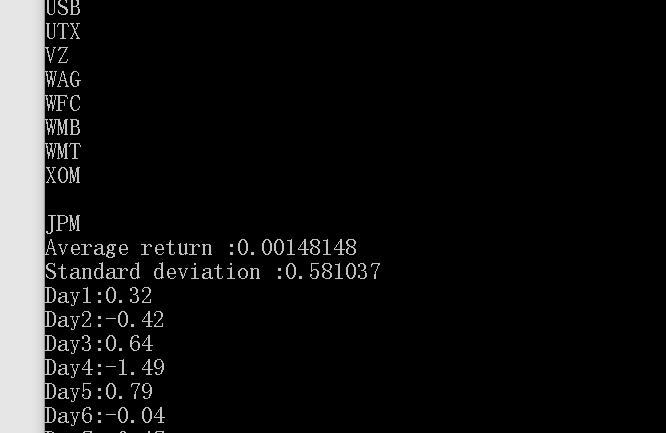




在主函数中将其他两个函数注释后运行test1函数，测试内容是每个asset的名称以及第1个asset的全部数据。

 (只截取部分)

修改成第51个asset输出也是正确的。

（只截取部分）