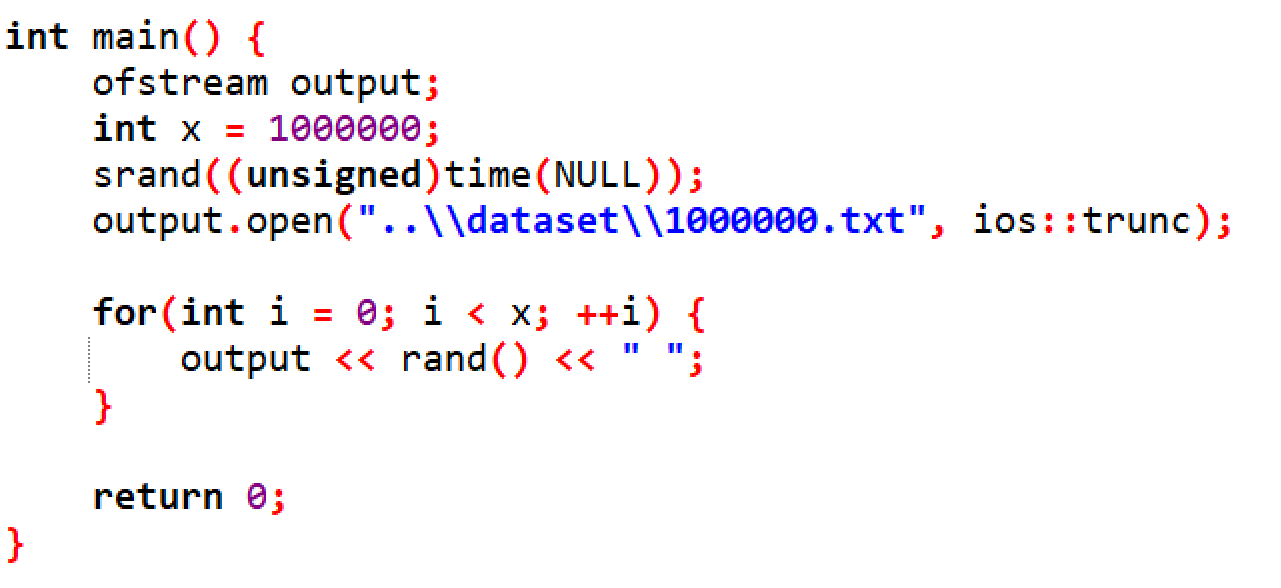
**【实验结果】**

|  |  |
| --- | --- |
| 数据规模 | 运行结果 |
| 100 |  |
| 1K |  |
| 10K |  |
| 100K |  |
| 1M |  |

**【实验分析】**

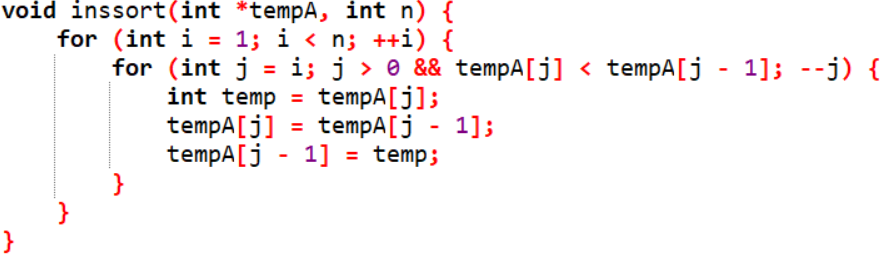
这是在英特尔酷睿i7-7700HK处理器上，运行Windows10操作系统时，插入排序算法和快速排序算法的运行时间比较。

数据来源为使用C++程序生成。下面给出代码截图：

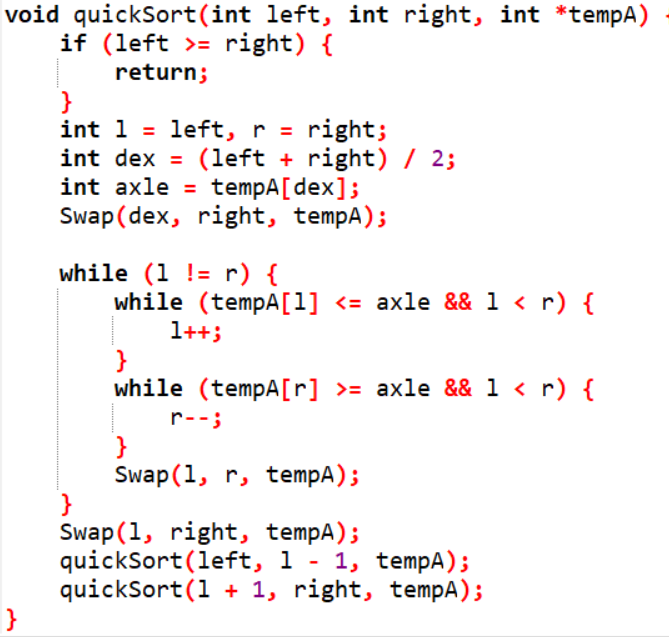


使用rand()函数生成随机数，并将其写入到文件中。注意使用了srand函数指定随机种子，生成不同序列的随机数。

插入排序：插入排序是一种时间复杂度为的排序算法。从实验结果中我们也可以看到当数据规模成十倍增加时，运行时间以百倍的速度增加。在数据规模达到100K时，就可以感受到明显的等待时间。



快速排序：快速排序是一种时间复杂度为的排序算法。从实验结果中我们也可以看到当数据规模成十倍增加时，运行时间以十倍的速度增加。在数据规模达到1M时，仍然感受不到明显的等待时间。



从两个算法比较来看，小样本数据差别不大；但是对于较大的实验数据，如：1M的数据规模，快速排序在极短的时间内得到结果，但插入排序则需要等候十几分钟，与的差距还是非常明显。

【**实验日志**】

2019年12月18日

完成随机数据的生成工作，学习了C++中rand函数的使用方法。知道了随机种子等概念。

2019年12月19日

完成排序代码的编写，在编写快速排序时出现了一点问题，首尾元素混乱，经过调试成功解决问题。完成本次实验。