## 9月7日

开始阅读《C++ Primer Plus 中文译版》

第一章，了解了一些有关C++的历史。以及编程步骤。

Q: 面对过程编程和面对对象编程有何区别？

## 9月9日

使用MircroVisualStudio2022创建了第一个程序。

学习了C++源代码的简单组成

#include<...>

int main()

{

statements;

return 0;

}

开始尝试level0.2

搜索if() ，else，while（）语句的用法。

用cin获取输入,if-else判断输入内容，while()重复，cout输出。

Q: return 0;为何是0？能不能是其它数字？0去了哪里？

函数名字一定得是main()吗？

## 9月10日

尝试写了一个猜数字的小程序。

运行良好直到输入了错误的数据类型（字符串），出现死循环。

开始四处寻求帮助

## 9月11日

通过AI工具解决了9月10号的bug。

错误的输入类型，导致输入流进入错误状态，从而跳过了接下来一切的cin语句，从而进入死循环。

通过cin.clear()和cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n')来清除输入流的错误状态。

并且加上cin.fail()检测，检测输入流是否出现错误。

发现cin的读取输入会在空格处截断，导致1 2 3 4 ... 100可以直接猜出答案！

使用函数反复调用来制造循环，重写了test1。

## 9月12日

学习了C++中的整型们

符号常量的定义方法（#define 或 const）

## 9月13

开始学习最为复杂的整型char类型

了解了转义字符以及用\u来表示特殊字符的方法

## 9月14日与15日

开始思考level0.3

开始大量使用互联网搜索。

了解到getline(),比cin >> 更好，不仅能读入空格，而且能自定义分界符（解析）。

count()可以计算字符串中’:’个数，以排除干扰行。

通过视频了解到fstream头文件，解决了文件的读入问题。

string数组解决了读入文档数据的临时存储问题。

发现getline只能作用在流上

开始形成初步的解决方案:文本文件→读入→一行行读取→count判断’:’个数，排除干扰行→流化→每一行再读取分隔→奇偶存储→流状态清除。得到将关键词放在偶数位，值放在奇数位的数组。

接下来可以着手搭建输入、检测和输出。

成功完成level0.3！

## 9月19日

开始完成level0.1

下载了Visual Studio Code，并根据视频的指导，布置了Git，并且登陆了自己自19年创建便一直在吃灰的GitHub账号。简单了解了它们的作用机制和关系。

安装了MinGW64,实现了用g++来编译程序。

上传了自己的test1与test2。

## 9月21日

将所有的level0聚合在一起写出了UltraTest，其中额外内置了一个简单的文本编辑器。过程中重新写了Test1与Test2，使用了continue;与break;语句使得程序更加灵活简单。

利用在解决level0.3时的文本分隔方法来对命令进行分割解析。

利用字符串流的特性，实现对字符串是否是数字（后来发现检测的是是字符串最前面的字符）的判断，用stoi()来将字符串转化成数字，从而解决了输入错误类型导致错误的问题。

开始进行对图书馆项目的编写。

## 9月22日到27日

了解了C风格的字符串。

在如何表示存储座位数据上卡住了。

后来想出的解决方案：id|floor|line|colum|reservationstatus1|2|3|4|5|6|7

加上二维数组（并不好，数据量太庞大，但是先实现才能谈优化）

通过chcp 65001控制台命令使其能够显示UTF-8编码的中文。

使用了大量的穷举。

解决了Login以及座位显示系统。

开始实现座位预约系统。