学习文档

在学习此之前，我只会 printf 与 scanf 函数，并且从来不会用 VS 等编译器。

Level0-2

学习文件的时候写出了自己引用文件的代码，可是总是运行错误，我无法理解，仔细比对也没有发现问题。

level1-1

明白了#include 的含义，调用库函数；

初次使用 scanf 会报错，原因是没有取消对某些函数的禁用；

学习了数据类型，整形，字符型，并学会 for 循环打印物品。

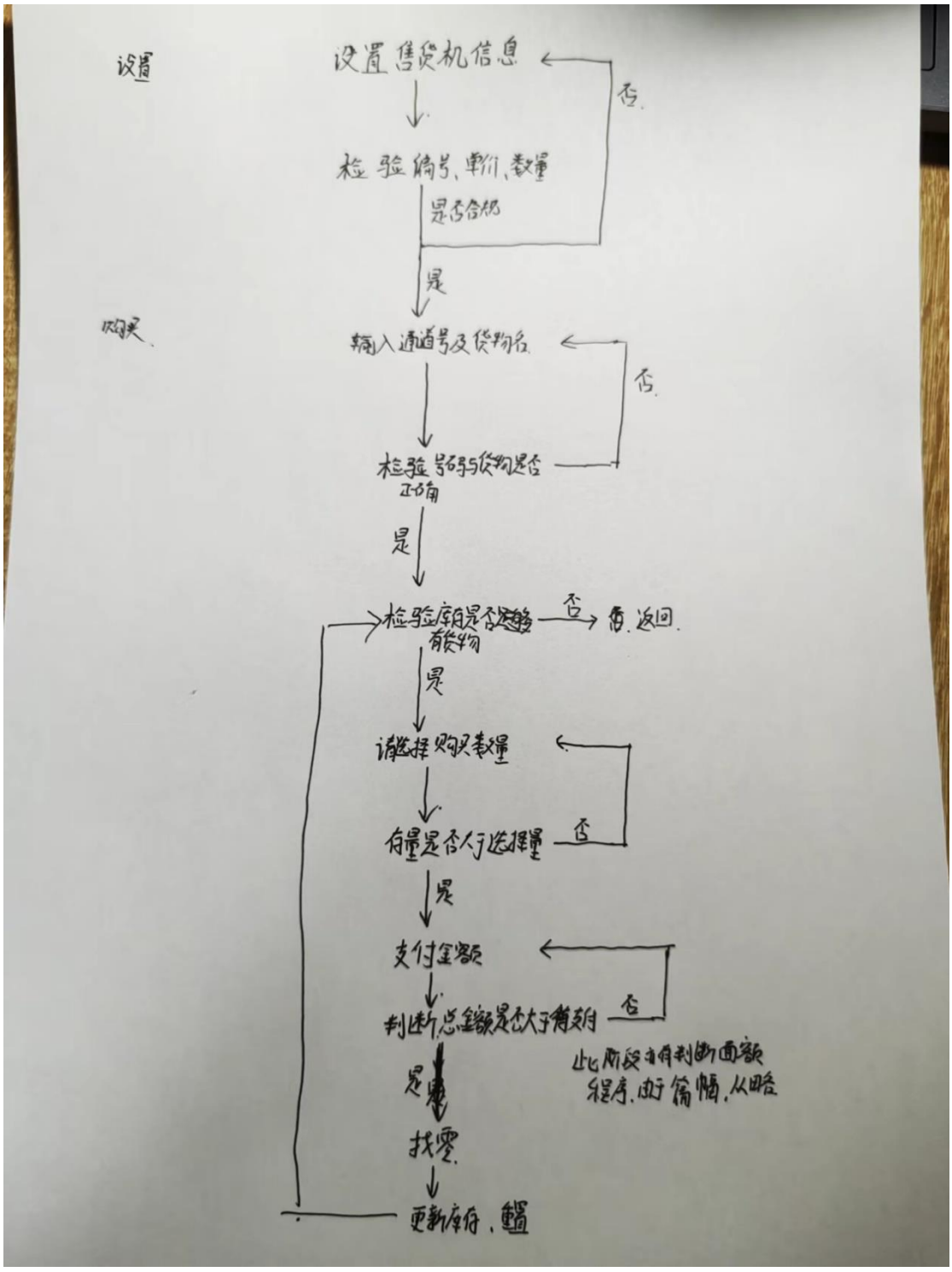
level1-2

此阶段更加注重对函数的命名一定得根据实际意义来，并常用大写形式；

学会了 while 语句，在总支付小于总价的条件下不断输入金额，并学会了逻辑“与”和逻辑“或”的使用。

level1-3

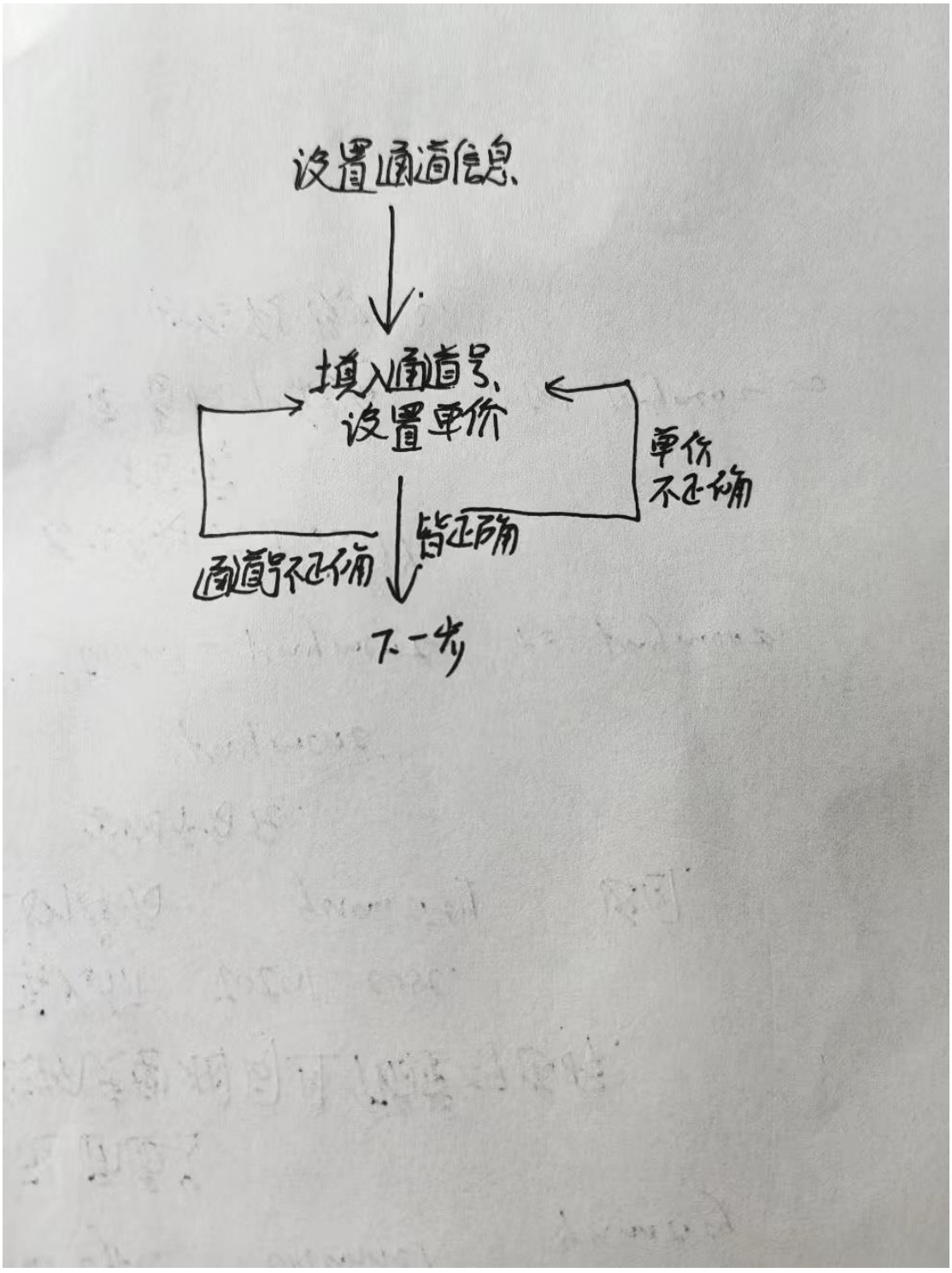
难度提升了，此过程养成了先思考程序逻辑过程的习惯，并在草稿纸上标注了逻辑过程（类似程序框图，不过高中没有系统学过，我只是采用自己理解的方式）；



学会了宏定义；

学会了在循环中嵌套循环。

此阶段难点在于正确的做出程序框图，由于这是我第一次自主设计程序框图，我无从下笔，甚至遇到了这样的问题



我没法用循环解决上述问题！后来我灵光一闪，可以先检验通道号，正确了再检验单价。

这个由零到一的过程是振奋人心的！

level2-1

难度提升了一个档次，在设计程序流程时我请教了好友及计科出身的姐姐。

画出了具有完整逻辑线的程序流程;

在好友及姐姐的介绍下，我上网查找并简单学习了以下函数（当然以下函数作用绝不止这些，还需要我逐步学习）:

了解了 fgets 函数读取字符串的作用;

初步了解了 strcspn 函数在修改字符串中某一字符的作用;

了解了 strcmp 函数比较字符串的作用;

了解了 sscanf 函数检验输入格式的作用。

level2-2

此阶段我遇到最大的问题便是“我到底该如何设计这个程序？”或言之“我到底如何将撤销操作加入 2-1 程序中？”

这个问题卡了我很久，以至于我花了很多时间却无济于事，我得承认，这一步，凭我自己无法想出来，我最初是想用 goto 语句，但是这样很难实现连续撤销，（况且我对此也不太熟悉），我再次向姐姐寻求帮助，在此过程中，我也了解到了国产 AI 文心一言，在不断询问，测试过程中，我了解到了可以使用数组记录操作（简单了解了“栈”的概念，并在 B 站上稍微学习了一下），并创立了撤销函数以及“行为”结构体，在程序中，每当用户选择一个商品时，都会将行为记录下来（这里面又暴露出了一个问题，就是数组总是以 0 作为第一个元素的下标，让我很头疼），这样我可以定义一个变量来记录操作次数，甚至可以规定连续撤销不超过 3 步!

按照我 2-1 的逻辑，用户只需要输入一次（不管输入什么，END 也好，BACK 也罢，设置物品也可）都会触发相应的程序，在不合适的时机输入 BACK，会提示输入错误，我在 2-1 的基础上改进了代码。下面的是我部分的学习记录。(图片较多就不一一展示了)

下标引用、函数调用和结构成员

[]　()　.　->

```
int arr[10]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    arr[3]=20;          // []为下标引用操作符. arr与3就是
                        // []的操作数. arr[3]=20是将数组中
                        // 的4號元值为20
```

同样也可以n=3代入

关键字.

所有局部变量前都应加上 auto 但凡所有局部变量都是auto类型的,

所以可以省略

break 跳出循环　　static 静态的.

自定义的类型.

enum 枚举　　typedef 类型重命名

union 联合体

struct. 结构体.　　~~voed~~ void 无类型　(函数返回类型. 参数)

volatile.

extern 声明外部符号的

register 寄存器

return 函数返回值

signed 有符号的.

unsigned 无符号的

/:

单目操作符：（只有一个操作数的操作符）

!	逻辑反操作	sizeof	（对类型不可以（），对变量可以（））操作数的类型长度（以字节为单位）
-	负值	~	对一个数的二进制按位取反
+	正值	--	前置、后置 --
&	取地址	++	前置、后置 ++
*	间接访问操作符（解引用操作符）	(类型)	强制类型转换

C语言中用0表示假，非0表示真。

例 int flag=0

```
if (!flag)
{ print("hehe\n");
}
```

⇒ hehe

int flag=2.

```
if (!flag)
{print("hehe\n")
}
```

⇒ ~~...~~ (空)