编码前工作；

刚开始自己更这网上的攻略安装了VMware 并在上面安装了Linux的ubuntu系统，然后开始配置环境，但是配完环境用wsl和VScode连接上之后用VScode（ubuntu）开发时总会出现找不到main函数入口，在网上搜索后改了mingw路径后来连include都识别不了了。然后做了很多无用功放弃了，之后尝试在虚拟机ubuntu终端用vim简单开发但是也会遇到各种错误我就放弃了，然后用window系统开发。

我跟着网上的课程了解到make工具是在makefile中声明目标：依赖项 然后要求进行的指令 还有伪目标我只是粗略地学了一下，然后再终端方面我学会了c语言编译四个步骤：

gcc -E main.c -o main.i

gcc -S main.i -o main.s

gcc -c main.c -o main.o

gcc mian.o -o main

还有就是通过gcc main.c -o main.exe

还学到了终端操作的md、rm、mkdir、touch、mv、cp的操作

leve-1.1实现说明：

eve-1-1主要通过for循环遍历命令行然后再用strcmp比较来实现

if实现判断

leve-1-2实现说明：

通过你叫终端输入字符串，fgets获取输入，打开关闭文件等函数实现功能

leve-1-2用到了缓冲区限制指令输入长度

还学到如何处理文件

其中学习并用到了以下内容：

strcmp strcmp(aoieue,aoieue) == 0(第三项加数字就是比两字符串前几位相等)

isspace((unsigned char)line[len - 1])  //判断是否为空格

strchr(line, '#')  //查找#在字符串中的位置 并返回该位置的指针

FILE\* input\_file = fopen(INPUT\_FILE, "r");  //打开文件 读模式 文件指针

fgets(buffer, BUFFER\_SIZE, input\_file)  //从文件中读取一行到buffer中

fputs(buffer, output\_file);  //将buffer中的内容写入到output\_file中

fputc('\n', output\_file); //fputs 不会自动添加换行符 需要手动添加

fclose(input\_file);  //关闭文件

fopen("Minimake\_cleared.mk", "w");  //写模式 如果文件不存在则创建 如果存在则清空原有内容

Leve1-3实现说明：

leve-1-3和leve-1-2用到的东西有很多重合

通过isspace实现空白的删除还有memmove移动字符然后就是文件操作

memmove 是把 str + i 位置的内容往前移动 i 个位置，len - i + 1 是为了把字符串结尾的 '\0' 也一起移动过去(一共移动的位数)

fprintf(stderr, "Line%d: :Missing colon in target definition\n", line\_number);  //标准错误输出

Leve2-1实现说明：

leve2-1比较难

我根据样例做了些比较简单但是有点错误的编码

其中用到了typedef struct 定义结构体

其中用到了typedef struct 定义结构体，然后用字符串处理方式将冒号分成两段在通过比较得到错误我并没有将依赖项全分出来，通过连表各项比较实现目标

其中学习并用到了以下：

strdup strdup(dependencies);  //字符串复制 并分配内存

perror("为 MakeRule 分配内存失败");  //打印错误信息

strstr(line, ":");  //查找字符串中第一次出现:的位置 并返回该位置的指针

strncpy(target, line, MAX\_NAME\_LEN);  //字符串复制 指定最大长度

strcspn(command, "\n")  //查找字符串中第一次出现\n的位置 并返回该位置的索引

fgets(command, sizeof(command), stdin)  //从标准输入读取一行到command中

strstr(src, sub)  //查找字符串中第一次出现sub的位置 并返回该位置的指针

memcpy(dst, src, copy\_len);  //内存复制 将src中的内容复制到dst中 指定复制的字节数