Lab2 Preprocessor

1. **说明**

该实验要求实现一个源代码预处理器(preprocessor)来实现代码文件中对宏定义指令(macro directives)的预编译。代码在进行编译之前，预处理器会将代码中的宏定义进行处理。常见的指令有：#include, #define, #undef, #ifdef, #ifndef, #if, #endif等等。

你可以使用g++ -E test1.cpp > output.cpp来在output.cpp中查看test1.cpp在进行预编译之后的代码。

在这个lab中，你需要实现一个预编译处理器（不需要处理C++标准库的include）来处理简单场景下的指令预编译。

1. **要求**
2. 实现：在lab中你需要将你的代码写在lab2.cpp中，lab2.cpp中提供了一个入口函数，你需要在这个入口位置实现你的代码。
3. 测试：在test文件夹中有两个cpp测试文件。test1.cpp很简单，可以帮助你在早期阶段debug。test2.cpp相对比较复杂，需要你仔细辨别各种宏指令处理的场景。
4. 运行：首先你需要编译lab2.cpp（使用c++11标准编译），并且运行编译后的文件。运行之后，会在test文件夹中生成两个test.out.cpp文件。然后在test文件夹中运行run\_tests.sh文件，在运行这个文件之前确保你在test目录中。
5. **评分标准**

* #include – 10%（不需要处理#include “iostream”）
* #define (check1 to check5)– 10%
* #undef – 10%
* #ifdef – 10%
* #else – 10%
* #ifndef – 10%
* #if – 10%
* #define function(PART 2) – 10%
* #define function(PART 3) – 10%（对#和##的处理各5%，对#和##的处理只要能通过测试文件即可）

**不存在内存泄露问题** - 10%

1. **提交**

将写好的lab2.cpp、test文件夹以及你的运行截图打包命名为【学号+姓名.**zip**】上传到ftp。