**【问题描述】**

编写一程序检查C源程序文件中{}、()等括号是否匹配,并输出第一个检测到的不匹配的括号及所对应括号所在的行号（程序中同一类括号只有一个不匹配）。

注意：

1.除了括号可能不匹配外，输入的C源程序无其它语法错误。

2.字符常量、字符串常量及注释中括号不应被处理，注释包括单行注释//和多行/\* \*/注释

3.字符常量和字符串常量中不包含转义字符\'和\"；

4.程序中出现有意义括号的个数不超过200个；

不匹配判断规则：

1.当检测的程序括号为'{'时，若其前序尚未匹配的括号为'('时，输出该'('左括号及所在行号；

2.当遇到一个不匹配的右括号')'或'}'时，输出该右括号及所在行号；

3.当程序处理完毕时，还存在不匹配的左括号时，输出该左括号及所在行号。

**【输入形式】**

打开当前目录下文件example.c，查询其括号是否匹配。该文件中每行字符数不超过200。

**【输出形式】**

若存在括号不匹配时，应输出首先能判断出现不匹配的括号及其所在的行号。当出现括号不匹配时，按下面要求输出相关信息：

without maching <x> at line <n>

其中<x>为‘{’, ‘}’， ‘(’, ‘)’等符号，<n>为该符号所在的行号。

若整个程序括号匹配，则按下面所示顺序输出括号匹配情况，中间没有空格。

(){(()){}}

【样例输入1】

若当前目录下输入文件example.c中内容如下：

#include<stdio.h>

int main(){

printf("{ hello world }\n"); // }

)

【样例输出1】

without maching ')' at line 4

【样例输入2】

若当前目录下输入文件example.c中内容如下：

#include<stdio.h>

int main(){

printf("{ hello world }d\n"); /\* }\*/

【样例输出2】

without maching '{' at line 2

【样例输入3】

若当前目录下输入文件example.c中内容如下：

#include<stdio.h>

int main(){

printf("{ hello world }d\n"); /\* }\*/

}

【样例输出3】

(){()}

【样例说明】

样例1：在注释部分和字符串中的括号不考虑，在将程序处理之后得到的括号序列是（）{（）），遇到右括号时与最近的左括号匹配，发现最后一个小括号和大括号不匹配。

样例2：处理之后的括号序列是（）{（），在最后缺少了右大括号，那么应该输出与之相对应的左括号不匹配。

**【评分标准】**

通过所有测试点得满分。