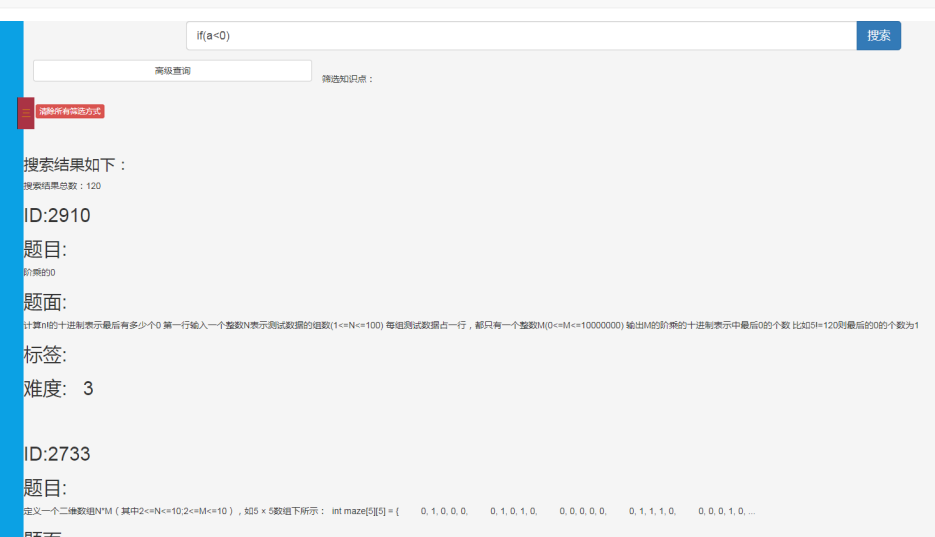
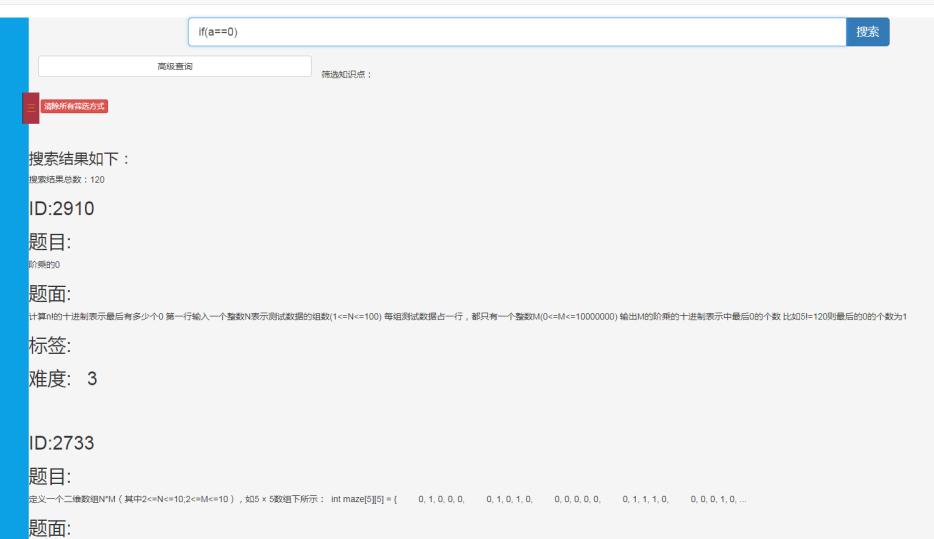
NO.1

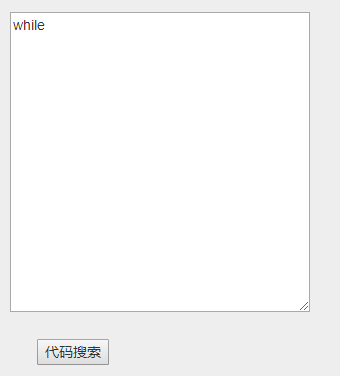
题面搜索中出现的匹配问题：



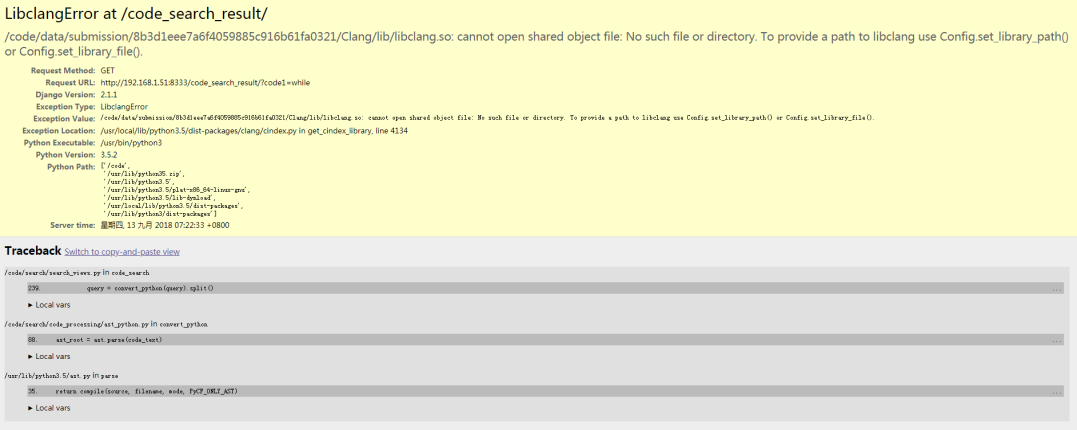
这个例子中if(a==0)和if(a<0)只匹配了0

NO.2

代码搜索中出现的无法匹配关键字的问题：

...

出现的结果都形似：



关键字目前都无法进行匹配

解答：这个错误信息是因为C++语法解析工具所需要的一个文件路径没找到，之前获取路径的方式在本地是没有问题的，现在我们优化了系统自动获取到这个文件路径的方式，使它在服务器端也可以正确获取。

NO.3

代码匹配中出现了无匹配项出现的搜索结果：

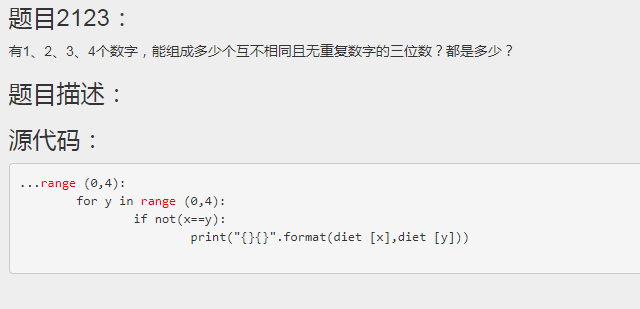


搜索range，出现第一条的源代码中就没有出现和range匹配的内容。

解答：这里我们进行了优化，在返回搜索结果时，我们会对结果进行再一次过滤，去除在字符串上完全没有匹配的结果。

NO.4

搜索range出现过的结果复制其中的条目再进行匹配时报错：



复制

for y in range (0,4):

出现的搜索结果为：



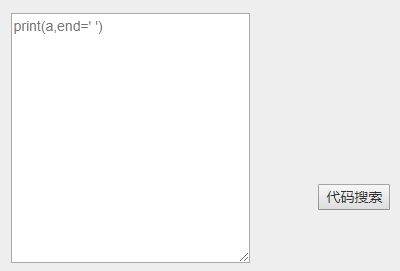
代码片段未能匹配

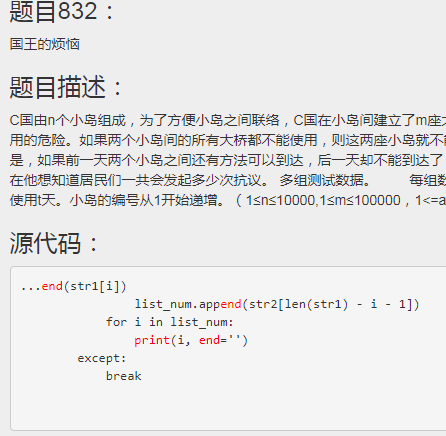
目前print的各种匹配是最成功的，print的匹配中可以进行较长代码片段匹配。

解答：这个错误信息是因为C++语法解析工具所需要的一个文件路径没找到，之前获取路径的方式在本地是没有问题的，现在我们优化了系统自动获取到这个文件路径的方式，使它在服务器端也可以正确获取。

NO.5

代码中特定表达式的匹配问题：





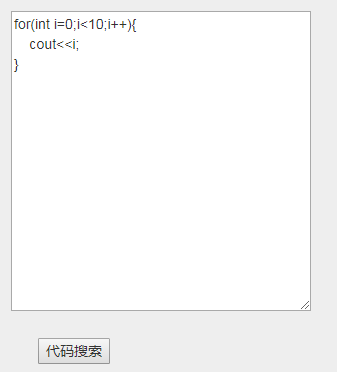
特定的表达式匹配，最后匹配出的结果将按内容匹配的结果进行了标红。

解答：这里是代码高亮部分的问题，由于我们的搜索是基于代码解析结果进行的，但是我们无法通过代码解析器反向的找到相对应的代码片段，所以我们的解决方法是通过查询代码和搜索结果代码之间的字符匹配进行高亮，这种字符匹配的方法就会带来一些小问题，比如上面的查询代码中包含end，搜索结果中的append也包含end，这个append中的end就也会高亮。

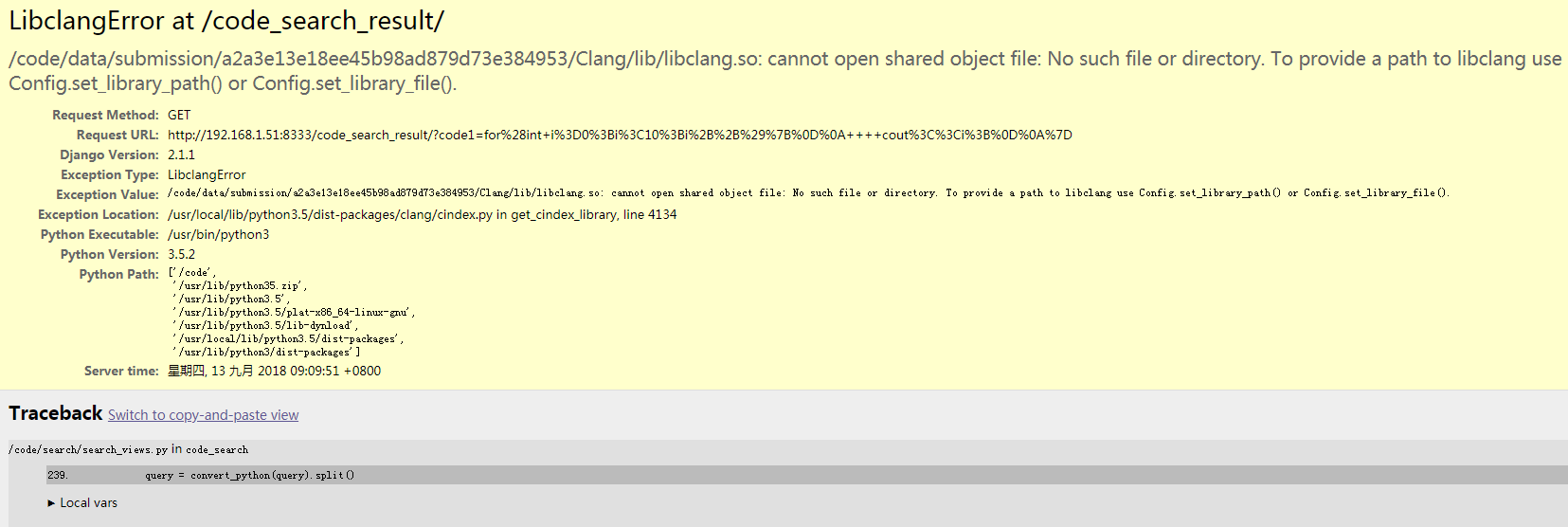
这个问题暂时还没有找到解决方法，因为这个高亮功能是django-haystack自带的，所以要改的话，可能要涉及到更改包内文件。

NO.6

代码匹配中还有代码片段的匹配未完成



上图为想要搜索的代码片段

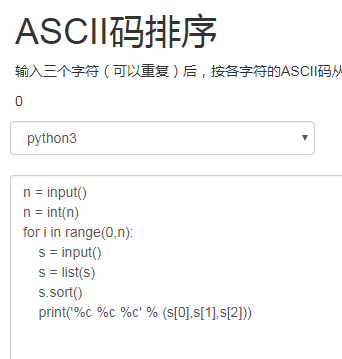


上图为未找到文件的报错信息

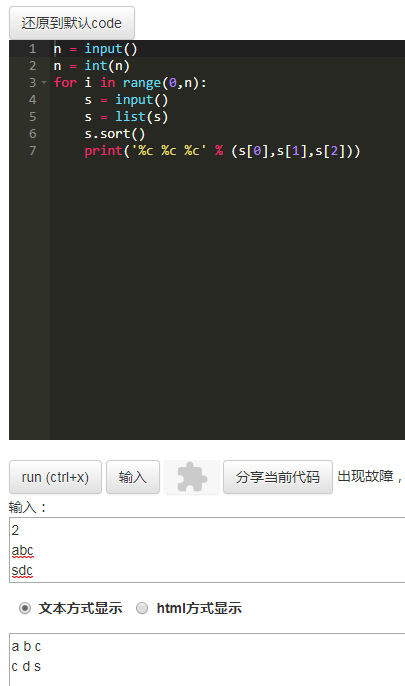
解答：这个错误信息是因为C++语法解析工具所需要的一个文件路径没找到，之前获取路径的方式在本地是没有问题的，现在我们优化了系统自动获取到这个文件路径的方式，使它在服务器端也可以正确获取。

NO.7

OJ系统判定问题：



上图为输入的代码



上图为这条代码在在线编辑器中运行出来的结果

而OJ系统却报类型错：

