实验二 标识符的识别

姓名：徐欣

班级：2017级计算机基地班

学号：320170917990

一、实验内容

【任务名称】标识符的识别
【任务性质】必做任务，分值30分。
【任务介绍】根据给定源语言的构词规则，从任意字符串中识别出所有的合法标识符。
【输入】字符串。
【输出】单词符号流，一行一个单词。

【题目】设计一个程序，从任意字符串中识别出所有可视为C语言“名字”的子串。注意：
（1）构词规则：以字母打头，后跟任意多个字母、数字的单词；长度不超过15；不区分大小写；把下划线视为第27个字母。
（2）关键字保留，即：语言定义中保留了某些单词用作关键字，程序员不可以将这些单词用作“名字“（变量名、常量名、函数名、标号名等等）。
要求：
（1）输入文件：input.txt，纯文本。内容是一段经过预处理的“干净的”源程序/片段，不考虑源程序中有错误的情况。
（2）输出文件：output.txt。内容是单词符号流，一行一个标识符，其顺序应该与源程序中的出现顺序完全一致。
（3）处理过程：要求在实验报告中说明对主要数据结构和算法的设计，包括：单词符号的数据结构和核心处理过程（如何识别标识符，如何挑出关键字等）。
（4）程序代码：要求以附件形式提供能编译运行的程序代码文件（包）。

二、实验设计

1. 伪代码

```
BEGIN
    WHILE(按行读入代码段) {
        跳过空行;
        跳过#include开头的头文件声明语句
        预处理：
            IF '字符常量' THEN 删除字符常量
            IF [...] THEN 删除[...]
            IF 字符串常量 THEN 删除字符串常量
    }

    用正则表达式匹配出符合要求的字符串
    从中去除关键字
    输出标识符
END
```

2. 相关技术说明

(1) 测试用例

- 测试用例过于”干净“，处理如下
  - 测试用例3中 *NFAState* buffer[MAX_STACK_LENGTH];* ，显然MAX_STACK_LENGTH是一个宏定义，但是没有#define，output.txt也没有当成标识符，所以直接把按在[]中的内容和[]删除
  - 因为没有#define，所以不把C语言中宏定义拿出来作为和关键字一样排除，对#开头的一行默认为#include...导入头文件，直接忽略
- 提供的测试用例1中的第3和第4行int后是两个Tab键当作空格，在实验时发现C++字符串无法识别，没有输入两个Tab空格，所以请用实验报告末的测试用例1测试（int + 1个空格 + age; int + 1个空格 + score）
- 测试用例3并不是“完全纯净"，其中最长的标识符长度为19，所以相应的正则表达式有所更改见下放说明

(2) 预处理

- 为了方便匹配标识符，将输入代码段中的空行、头文件包含语句、字符常量、[...]和字符串常量这些不可能出现标识符的地方在输入时直接过滤
- 后三者用substr截取

(3) 正则表达式

- 以字母打头，后跟任意多个字母、数字的单词；长度不超过15；不区分大小写； 把下划线视为第27个字母。

```
[a-zA-Z_][a-z0-9A-Z_]{0,14}
```

  - 因为测试用例3中标识符最长为19，所以改成了

```
[a-zA-Z_][a-z0-9A-Z_]{0,18}
```

- 用C++中的regex模块处理正则表达式
  - 创建正则表达式对象

```
regex e("[a-zA-Z_][a-z0-9A-Z_]{0,18}", regex_constants::icase);
```

  - 在一个字符串中找到匹配的子串并存储

```
while(regex_search(ans[i], m, e)) {
    for(auto x:m) res.push_back(x);
    ans[i] = m.suffix().str();
}
```

(4) 去除C语言关键字

- 将C语言32个关键字存放在set中（因为C++STL中的set默认会排序，为了提高效率用了unordered_set）

- 利用set中的find方法，将匹配出的字符串中是关键字的剔除

三、实验环境

- 语言：C++
- 编译环境：Code Blocks 17.12 自带MinGW中的g++，需要在编译器设置中勾选C++14
- 3个测试用例输入如下，复制粘贴最后手动加上**Ctrl+Z**——文件结束符

  测试用例1：

```
struct{
    char[10] name;
    int age;
    int score;
}student;

int main(){
    struct student stu;
    int stu_age = 22;

    stu.name = "ellise";
    stu.age = stu_age;
    stu.score = 99;

    return 1;
}
```

  测试用例2：

```
#include <stdio.h>

int main()
{
    printf("Hello, World! \n");
    return 0;
}
```

  测试用例3：

```
typedef struct _NFAStateStack
{
    NFAState* buffer[MAX_STACK_LENGTH];
    int top;
}NFAStateStack;


void InitNFAStateStack(NFAStateStack* pS);
void PushNFAState(NFAStateStack* pS, NFAState* Elem);
NFAState* PopNFAState(NFAStateStack* pS);
int NFAStateStackEmpty(NFAStateStack* pS);
```