Null 语法分析实验报告

封雨希 1511485，吴珊珊 1511501

实验目的

利用flex和bison进行语法分析+ 语义分析初级+ 类型检查，完成一个语法分析器

实验环境

ubuntu 16.04 64位 + flex + bison

运行

在终端下进入目录Compiler。

用`make`运行程序。命令行输入

```
make
```

即可使用makefile中确定好的编译步骤编译程序，结果输出为parser。然后即可以运行以下命令将testinput1.c作为输入重定向程序中

```
./parser < testinput1.c
```

如要改变输入文件，命令中的testinput1.c改为相应的文件路径即可。

实现

词法分析器

在作业一的基础上，进一步优化词法分析器。我们在`tokens.l`文件中运行Flex，生成`tokens.cpp`文件。这个cpp文件将和我们的语法分析器一起被编译，并提供可以识别所有这些记号的`yylex()`函数。

语法分析器

语法分析器的最终结果是生成一棵抽象语法树，所以我们的工作就是解决如何将结点插入语法树中的问题。

首先在插入字符串到抽象语法树之前，我们需要创建一个数据结构`Node`来存放，其定义在`node.h`文件中。

接着我们在bison将要编译的parse.y文件中定义识别每个语法后的一系列操作，它们在被识别之后执行。执行顺序的按照由叶结点到根结点的顺序递归地完成的，其中每个非终结符最终都将被合并成一棵树。

人员分工

- 封雨希：基础要求
- 吴珊珊：基础要求、测试及编写文档

支持

自定义语法部分范围及评分标准					
组成	要求类型	具体要求	是否必须	得分	12.5
数据类型	int,char	支持上述类型的运算，定义等操作	是	1	1
	float,double	支持上述类型的运算，定义等操作	否	1	1
基本语句	赋值语句	支持C语言中的赋值语句	是	0.5	0.5
	if语句	支持C语言中的if语句	是	1	1
	while语句	支持C语言中的while语句	否	1	1
	for语句	支持C语言中的for语句	否	1	1
	完成全部以上四种	无	否	1	1
基本运算	四则运算	支持加减乘除四则运算	是	0.5	0.5
	其他算术运算	支持取模运算以及位运算	否	1	1
	关系运算	支持>=,<=,==,>.<	是	0.5	0.5
	逻辑运算	支持&&,||,!	否	1	1
	支持全部以上四种	无	否	0.5	0.5
代码块部分	复合语句	支持{}	是	1	1
	函数	支持c语言中的函数	否	1	
	结构体	支持c语言中的结构体	否	1	
	类	支持c++中的类(与结构体不叠加)	否	2	
注释	单行注释	//	否	0.5	0.5
	多行注释	/* */	否	0.5	0.5
其他功能	输入输出语句	可简单定义为read，wirte	是	0.5	0.5
	数组	支持数组(只支持一维数组0.5分)	否	1	
	函数重载	支持函数重载	否	1	
	自动类型auto	支持自动数据类型auto	否	0.5	
	指针	支持指针操作	否	1	
	继承	支持类的继承	否	1	
	虚函数	支持虚函数	否	1	
	其他创新功能	C++14支持的功能	否	酌情	
超出满分12分的部分，按照完成情况分配到个人，个人上限为15分					