实验4 语法分析

实验4的任务是在实验3的词法分析程序基础上，编写一个程序，对使用SysY语言书写的源代码进行语法分析，并打印分析结果。实验的实现方式可以手工编写程序，也可以采用词法分析工具GNU Flex和语法分析工具Bison等。

需要注意的是，由于在后面的实验中还会用到本次实验已完成的代码，因此，建议保持良好的代码风格，系统地设计代码结构和各模块之间的接口。

4.1 实验要求

程序要能够查出SysY源代码中可能包含的词法错误和语法错误:

词法错误(错误类型代码为A)：出现SysY词法中未定义的字符以及任何不符合SysY词法单元定义的字符。

语法错误(错误类型代码为B).

程序在输出错误提示信息时，需要输出具体的错误类型、出错的位置（源程序的行号）以及相关的说明文字。

4.2 输入格式

程序输入是一个包含SysY源代码的文本文件，程序需要能够接受一个输入文件名作为参数。例如：假设程序名为ss, 输入文件名为test1.sy，程序和输入文件都在当前目录下，那么在Linux命令行下运行 ./ss test1.sy即可获得test1.sy作为输入文件的输出结果。

4.3 输出格式

实验一要求通过标准输出打印程序的运行结果（也可以单独输出到文件，格式保持一致）。对于那些包含词法错误的输入文件，只要输出相关的词法错误信息即可（如果你能有出错恢复策略，把所有可能的错误找出来，更好）。要求输出的错误信息包括错误类型、出错的行号以及说明文字，格式如下：

```
Error type [错误类型] at line [行号] : [说明文字]
```

说明文字的内容不做具体要求，但是错误类型和行号要正确。**假设输入文件中可能包含一个或者多个错误，但同一行最多只有一个错误**。每一条错误信息在输出中单独占一行。

在实验一中，对于没有任何词法错误的输入文件，我们要求程序要将识别到的词法单元列表按顺序输出打印，每行代表一个词法单元的二元组:`<词法单元类型，属性值>`。词法单元类型编码方式暂时不做统一要求，可以先自己设计。在本次实验中，如果**不存在任何词法或语法错误**的输入文件，程序要将构造好的分析树按照先序遍历的方式打印每一个节点的信息，这些信息包括:

1. 如果当前节点是一个语法单元，且该语法单元没有产生ε，则打印该语法单元的名称，以及它在输入文件中的行号(行号用括号括起来，并且与语法单元之间用空格隔开)。 语法单元在输入文件中的行号是指该语法单元产生的第一个单词在输入文件中的行号(从词法分析的时候获得)
2. 如果当前节点是一个语法单元并且该语法单元产生了ε，则无需打印该语法单元的信息（只需打印名称)
3. 如果当前节点是一个词法单元，则只需要打印该词法单元的名称，而无需打印该词法单元的行号

   a. 如果当前节点是词法单元ID(标识符)，则要求额外打印该标识符所对应的词素(符号串)

b. 如果当前节点是词法单元类型关键词TYPE(INTTK或FLOATTK或VOID)，则要求额外打印说明该类型是int还是float(如果你做了float类型)

c. 如果当前节点的词法单元是常数(INTCON或FLOATCON)，则要求打印十进制的值(对应词法分析任务要求)

d. 词法单元所额外打印的信息与词法单元之间以一个冒号和空格隔开

词法单元编码方式以各小组在词法分析阶段定义的一致

1.4 提交要求

实验二要求提交如下内容：

1. 可被正确编译执行的源程序（使用工具的话，需要提供工具源码Flex/Bison和修改后的C代码，其他语言实现则需要说明配置环境）
2. 一份实验报告的PDF文件，内容包括：
   - [ ] 程序实现的主要功能，简要说明怎么实现的。
   - [ ] 程序如何进行编译？ 特别是采用冷门语言编写的代码
   - [ ] 实验报告总长度不要超过4页。重点描述的是自己程序的亮点和不一样的地方，对于相对简单的内容可以不提或简单提一下，杜绝大段粘贴源码。实验报告字体最小字号是五号字。

4.5 样例

例4.1 输入(行号是标识需要，并非样例输入的一部分，下同)

```
1  int main()
2  {
3    int i = 1;
4    int j = ~1;
5    return 0;
6  }
```

输出: 该程序存在词法错误。第4行中的字符“~”没有在SysY词法中定义过，因此程序需要输出以下错误信息

```
1  Error type A at Line 4: Invalid character "~"
```

例4.2 输入

```
1  int main()
2  {
3    int a[10][12];
4    int i;
5    a[5, 3]=5;
6    if (a[1][2]==0)
7      i=1
8    else
9        i=0;
10 }
```

输出：该程序存在两处语法错误，一是二维数组的正确访问方式是a[5][3]而非a[5,3]，二是第7行最后少了分号。因此程序需要输出两行错误提示信息

```
Error type B at Line 5: Missing "]".
Error type B at Line 7: Missing ";".
```

例4.3 输入

```
int inc()
{
    int i;
    i = i+1;
}
```

输出：这个程序非常简单，也没有任何词法错误或语法错误。程序只需要输出分析树结构即可，词法单元类型编码可自定义，语法单元名称根据SysY文法说明，如果经过修改，用自己修改过后的文法符号名称，保持一致即可

```
CompUnit (1)
  FuncDef (1)
    FuncType (1)
      Type: int
    Ident: inc
    LPARENT
    RPARENT
    Block (2)
      LBRACE
      BlockItem (3)
        Decl (3)
          VarDecl (3)
            BType (3)
              Type: int
            VarDef (3)
              Ident: i
            SEMICN
      BlockItem (4)
        Stmt (4)
          LVal (4)
            Ident: i
          ASSIGN
          Exp (4)
            AddExp (4)
              AddExp (4)
                MulExp (4)
                  PrimaryExp (4)
                    Lval (4)
                      Ident: i
              PLUS:+
              MulExp (4)
```

```
                PrimaryExp (4)
                  Number (4)
                    INTCON: 1
            SEMICN
      RBRACE
```

例4.4 输入

```
int main()
{
  int i= 0123;
  int j= 0x3F;
}
```

输出：该程序涉及到常数的八进制和十六进制，程序需要识别出常数对应的值，并在词法单元中给予呈现

```
CompUnit (1)
  FuncDef (1)
    FuncType (1)
      Type: int
    Ident: main
    LPARENT
    RPARENT
    Block (2)
      LBRACE
      BlockItem (3)
        Decl (3)
          VarDecl (3)
            BType (3)
              Type: int
            VarDef (3)
              Ident: i
              ASSIGN
              InitVal (3)
                Exp (3)
                  AddExp (3)
                    MulExp (3)
                      PrimaryExp (3)
                        Number (3)
                          INTCON: 83
            SEMICN
      BlockItem (4)
        Decl (4)
          VarDecl (4)
            BType (4)
              Type: int
            VarDef (4)
              Ident: j
              ASSIGN
              InitVal (4)
                Exp (4)
```

```
                    AddExp (4)
                      MulExp (4)
                        PrimaryExp (4)
                          Number (4)
                            INTCON: 63
            SEMICN
      RBRACE
```

如果上述程序中的常数分别改为“09”和“0x3G"，程序应该能分别识别出八进制数和十六进制数的错误，并打印相应的错误信息

```
Error type A at line 3: Illegal octal number '09'
Error type A at line 4: Illegal hexadecimal number '0x3G'
```

4.6 实验指导

词法分析和语法分析两部分内容是编译器当中被自动化得最好的部分。即使没有任何理论基础，通过工具也能短时间内做出很好的词法分析程序。但是并不是说这部分的理论基础不重要，恰恰相反，这个部分也可以认为是计算机理论在工程实践中最成功的的应用之一，对它的介绍也是编译原理理论课程中的重点。【具体介绍内容查看词法分析和语法分析工具指导相关文件，也可以自己查阅相关教程】