

学号：19373075

姓名：许天识

思路方法

考虑到lab02中part3其实是part4的一个子集，因此一并实现。经观察，发现文法存在左递归，原有的递归下降法失效，因此采用比较适合做表达式计算的算符优先法。

总体流程

本次实验与lab01相比，仅仅在非终结符Exp部分不一样，因此可以沿用先前的框架，添加上利用算符优先法对Exp表达式的识别，其中需要用到操作数栈和操作符栈两个数据结构。表达式识别的流程描述如下：

```
操作数栈：stackNum,操作符栈：stackOp
报错函数:error()
1.不断读入token，直到token读完或读到了";"
2.判断token是否是属于表达式中的字符，不是则error()，是则到(3)
3.判断token是否是数字，是则执行stackNum.push(token)，否则到(4)
4.判断操作符栈顶元素与当前token的优先级关系：
    1) 小于：执行stackOp.push(token)
    2) 等于：此时必定是"()",判断stackNum是否有元素，若无则error()，有则执行stackOp.pop()
    将 "("弹出，读入下一个token.
    3) 大于：执行规约操作
    4) 不存在关系：error()
    转1
5.循环退出，判断token是否等于";"，否则error()，是则继续进行规约，直到stackOp为空，此时判断
stackNum里面数字是否唯一，是则表达式识别完成，所剩数字即为表达式的值，否则error()
```

关键问题解决方案

1.区分单目运算符与双目运算符

"+"与"-"必须区分单目和双目，才能构建出算符优先矩阵，否则构建出来的会有冲突(血泪教训...)，经过观察不难得出：凡是前面的token是数字或者右括号的，是双目运算符，否则是单目。

2.算符优先矩阵的构建

采用纸上先手推出来，再打表的方法....