

# Lab1\_answer

## PB21111686\_赵卓

### T1

`\w+([-+.]w+)@w+([-.]w+).w+([-.]w+)*` 正则表达式匹配的字符串的含义是什么？

- 分析该正则表达式可知，具体含义可以表示电子邮箱。

### T2

匹配 HTML 注释：编写一个正则表达式，可以匹配 HTML 中的注释，例如 `<!-- This is a comment -->`。

- `\D+\s(\w+\s)*\D+` 即为可与 HTML 注释匹配的一个正则表达式。

### T3

如果存在同时以下规则和动作，对于字符串 `+=`，哪条规则会被触发，并尝试解释理由。

```
%%  
\+ { return ADD; }  
= { return ASSIGN; }  
\+= { return ASSIGNADD; }  
%%
```

- 第三条规则会触发，因为 Flex 匹配会从长规则到短规则依次匹配，此时第三条规则最长且与字符串 `+=` 匹配。

### T4

如果存在同时以下规则和动作，对于字符串 `ABC`，哪条规则会被触发，并尝试解释理由。

```
%%  
ABC { return 1; }  
[a-zA-Z]+ {return 2; }  
%%
```

- 第一条规则会触发，因为第一条和第二条规则都和 `ABC` 匹配时，Flex 会选择顺序在前的规则。

## T5

如果存在同时以下规则和动作，对于字符串 ABC，哪条规则会被触发，并尝试解释理由。

```
%%  
[a-zA-Z]+ {return 2; }  
ABC { return 1; }  
%%
```

- 第一条规则会触发，因为第一条和第二条规则都和 ABC 匹配时，Flex会选择顺序在前的规则。

## T6

上述计算器例子的文法中存在左递归，为什么 bison 可以处理？

- 因为bison是使用LALR将文法转为解析器的，LALR使用了前看符号，所以通过前看符号可以解决左递归文法出现的冲突。

## T7

能否修改计算器例子的文法，使得它支持除数 0 规避功能？

```
[0-9]+|[0-9]+\.[0-9]*|[0-9]*\.[0-9]+ { yy1val.num = atof(yytext); return NUMBER; }
```

- 词法分析器在读到非终结符 NUMBER 时，先判断 yytext 获取到的值是否为0，不为0才将它的语义值压入到 yy1val.num 中，否则不将其传到语法分析器中,修改之后，若除数为0，则直接报错，支持除数0规避功能。