



实验1 测试用例

戴新宇老师班

提交说明

- 地址: <ftp://114.212.190.181:31>
- 用户名和密码: upload
- 格式: 学号命名的压缩包 (*zip/rar*)
- 内容:
 - 源程序(*ex1.l, ex1.y; 额外的.c文件*)
 - 可执行程序(命名为*parser*)
 - 报告PDF(*完成的功能点, 编译步骤, 实现方法, 结点的数据结构表示; 不超过3页*)
- 备注: 可重新提交, 加后缀 **_02, _03**

- 总的要求

- 错误类型 + 错误行号; 不能漏报和误报
- 有错,不能打印语法树; 无错, 格式打印语法树

- 必做测试

- 测试1.1: 词法错误
- 测试2.1~2.6: 语法错误 (一处或者多处)
- 测试2.A~2.C: 无错, 格式打印语法树

- 选做测试

- 测试3: 进制和科学计数法
- 测试4: 两种风格的注释

测试1.1：词法错误

输入：

```
1. int main(){  
2.     int a;  
3.     a = a ^ 2;  
4. }
```

输出：

Error **type 1** at **line 3** : unknown char : ^

测试2.1 程序结构

输入：

```
1. int main(  
2. {  
3.     int nlp;  
4. }
```

输出：

Error type 2 at line 2

备注： *line 1 亦可*

测试2.2 变量定义

输入：

```
1. int main(){  
2.     int i;  
3.     i = 1;  
4.     int j;  
5. }
```

输出：

Error type 2 at line 4

测试2.3 标识符

输入：

```
1. int main(){  
2.     int 911nlp;  
3. }
```

输出：

Error type 2 at line 2

测试2.4 分号

输入：

```
1. int main(){  
2.     int nlp  
3. }
```

输出：

Error type 2 at line 3

备注： *line 2 亦可*

测试2.5 二维数组

输入：

```
1. int main(){  
2.     int a[1,2];  
3.     int c[1][2];  
4.     int b[3][4.0];  
5. }
```

输出：

Error type 2 at line 2

Error type 2 at line 4

测试2.6 数据类型

输入：

```
1. struct complex{  
2.     float r;  
3.     double i;  
4. };  
5. int main(){  
6.     complex c;  
7.     c.r = 1;  
8.     c.i = 0;  
9. }
```

输出：

Error type 2 at line 3

Error type 2 at line 6

备注: 多报错 *line 7,8* 亦可

测试2.A 基本功能

1. `int main() { }`

输出： 格式打印语法树

测试2.B 数组翻译

```
1. int main(){  
2.     int a[1][2];  
3.     int i = 0;  
4.     while( i < 2){  
5.         a[0][i] = i;  
6.         a[0][i] = i + 1;  
7.     }  
8. }
```

输出： 格式打印语法树

测试2.C 结构体翻译

```
1. struct complex{  
2.     float r,i;  
3. };  
4. int main(){  
5.     struct complex c;  
6.     c.r = 1;  
7.     c.i = 0;  
8. }
```

输出： 格式打印语法树

测试3.1 八进制

```
1. int main(){  
2.     int oct = 09;  
3. }
```

输出

Error type 2 at line 2

备注：仅针对选做该功能的同学

测试3.2 十六进制

```
1. int main(){  
2.     int hex = 0xG;  
3. }
```

输出

Error type 2 at line 2

备注：仅针对选做该功能的同学

测试3.3 科学计数法

```
1. int main(){  
2.     float a = 1.1e1.1  
3. }
```

输出

Error type 2 at line 2

备注：仅针对选做该功能的同学

测试3.4 科学计数法

```
1. int main(){  
2.     float a = .e1  
3. }
```

输出

Error type 2 at line 2

备注：仅针对选做该功能的同学

测试3.A 八进制与十六进制

```
1. int main(){  
2.     int oct = 010;  
3.     int hex = 0x1a;  
4.     int i = 10;  
5. }
```

- 输出
 - ✓ 选做了的打印语法树
 - ✓ 没做的同学要报错, 可以只报第一个错
 - ✓ 不转换为十进制亦可

测试3.B 科学计数法

```
1. int main(){  
2.     float a = 1.1e+1;  
3.     float d = 01.1E-1;  
4.     float b = .1E-1;  
5.     float c = 1.E+1;  
6. }
```

- 输出
 - ✓ 选做了的打印语法树
 - ✓ 没做的同学要报错，可以只报第一个错

测试4.1 注释

```
1. int main(){  
2.     float f = 1.0;    /* float number *  
3. }
```

备注：仅针对选做该功能的同学

- 针对未终止的注释进行测试：**From** 许畅老师班
- 如果打印了语法树，或者程序异常终止、死循环无法退出等，则该用例不得分。
- 不限定错误类型以及提示方式，但是出错位置必须限定在2行或者以后的位置；直接提示“未终止的注释”也可以。

测试4.A 注释

```
1. int main(){  
2.     // this is a single line comment  
3.     int i = 1;     //integer  
4.     /*  
5.         this is a  
6.         multiple lines comment  
7.     */  
8.     float f = 1.0; //float number  
9. }
```

- ✓ 选做了的打印语法树
- ✓ 没做的同学要报错，可以只报第一个错