

所有文件都放于lex\_c文件夹下，初始测试文件名为test.txt

test中代码：

```
1  #include<stdio.h>
2  int main() {
3      int a = 0xF1;
4      int x = 20;
5      float b = 20.5678;
6      int 12a = 0xw2;
7      if (a < b) {
8          a++;
9          b = b + 2.0;
10     }
11
12     string a = "abc\p";
13     //hello
14     char s = '\n';
15     return 0;
16 }
17
```

测试结果

```
1  Results of lexical analysis of C language programs
2
3  Preprocessor Line: #include<stdio.h>      token_id: 85
4  Keyword:int          token_id: 1
5  Identifier:main      token_id: 33
6  Operator: (  token_id: 78
7  Operator: )  token_id: 79
8  Operator: {  token_id: 80
9  Keyword:int          token_id: 1
10 Identifier:a          token_id: 33
11 Operator: =  token_id: 52
12 Unsigned xdigit: 0xF1 token_id: 34
13 Operator: ;  token_id: 77
14 Keyword:int          token_id: 1
15 Identifier:x          token_id: 33
16 Operator: =  token_id: 52
17 Unsigned integer: 20  token_id: 34
18 Operator: ;  token_id: 77
19 Keyword:float        token_id: 2
20 Identifier:b          token_id: 33
21 Operator: =  token_id: 52
22 Float: 20.5678  token_id: 35
23 Operator: ;  token_id: 77
24 Keyword:int          token_id: 1
25 Error:12a            Line: 5      Error id: 3
26 Operator: =  token_id: 52
27 Error:0xw2           Line: 5      Error id: 3
28 Operator: ;  token_id: 77
29 Keyword:if           token_id: 25
30 Operator: (  token_id: 78
```

```

31 Identifier:a          token_id: 33
32 Operator: <  token_id: 46
33 Identifier:b          token_id: 33
34 Operator: )  token_id: 79
35 Operator: {  token_id: 80
36 Identifier:a          token_id: 33
37 Operator: ++  token_id: 63
38 Operator: ;  token_id: 77
39 Identifier:b          token_id: 33
40 Operator: =  token_id: 52
41 Identifier:b          token_id: 33
42 Operator: +  token_id: 38
43 Float: 2.0  token_id: 35
44 Operator: ;  token_id: 77
45 Operator: }  token_id: 81
46 Identifier:string     token_id: 33
47 Identifier:a          token_id: 33
48 Operator: =  token_id: 52
49 Operator: "  token_id: 86
50 Error: Invalid Escape Sequence -- \p      Line: 11      Error id: 8
51 Constant String: "abc\p"  token_id: 37
52 Operator: "  token_id: 86
53 Operator: ;  token_id: 77
54 Comment: hello
55
56 Keyword:char          token_id: 4
57 Identifier:s          token_id: 33
58 Operator: =  token_id: 52
59 Operator: '  token_id: 87
60 Constant Char: \n  token_id: 36
61 Operator: '  token_id: 87
62 Operator: ;  token_id: 77
63 Keyword:return        token_id: 21
64 Unsigned integer: 0  token_id: 34
65 Operator: ;  token_id: 77
66 Operator: }  token_id: 81
67
68
69 Line: 15
70 Preprocessor Line: 1
71 key_word: 8
72 operator: 33
73 error: 2

```

这是测试的结果，左边是识别的token类型和内容，右边是对应的token\_id。

在错误时会出现Error字样，并标明出现错误的位置和错误的种类。

最下方是词法分析的统计结果，记录了一些主要token出现的次数。