National Sun Yat-sen University

國　　立　　中　　山　　大　　學

DESIGN AND IMPLEMENTATION OF C O M P I L E R

編　　譯　　器　　製　　作

Lex 報告

授課教師：張玉盈

系級:資訊工程學系 110 級

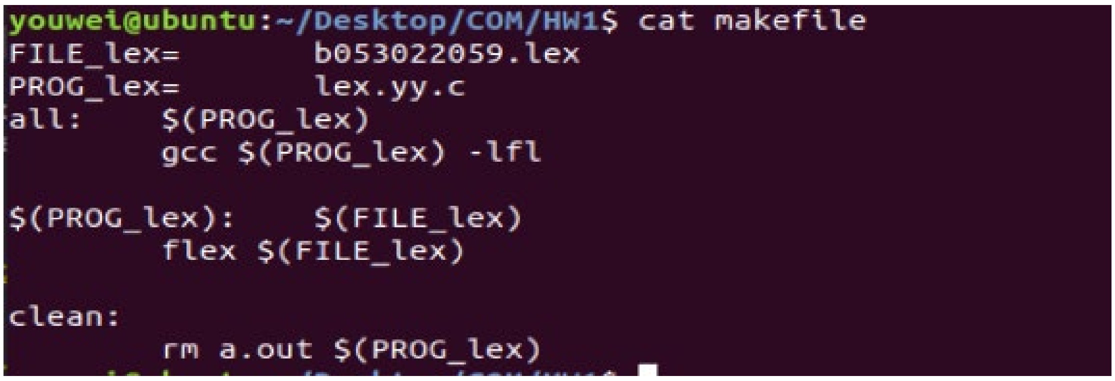
學生:B053022059 蔣有為

日期：109.4.17

1. Lex 版本: flex 2.6.4

2. 作業平台: Linux ubuntu 5.3.0-42-generic #34~18.04.1-Ubuntu SMP Fri Feb 28 13:42:26 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux

3. 執行方式:



```
youwei@ubuntu:~/Desktop/COM/HW1$ cat makefile
FILE_lex=       b053022059.lex
PROG_lex=       lex.yy.c
all:    $(PROG_lex)
        gcc $(PROG_lex) -lfl

$(PROG_lex):    $(FILE_lex)
        flex $(FILE_lex)

clean:
        rm a.out $(PROG_lex)
```

4. 如何處理這份規格書上的問題

(1). 保留字 (Reserved_word)

Pascal 是 case insensitive 。 例如：保留字 program、ProGram 和 PROGRAM 是一樣的。

因為是保留字，我選擇放在 rlues 中相當上面的位置，給予較高的優先權。

其中有提到，要針對保留字做篩選，並且是不分大小寫，然而許多的保留字在輸入的時候，會有不方便輸入且看不清楚，因此直接寫了一個 C 程式，直接輸出想要的保留字表示法列表。

(2). 識別字 (Identifiers)

1. 字數最長到 15 個字元。

先使用 int str_l = strlen(yytext);抓到的 token 的長度

並且　if(str_l > 15) 等判斷句去檢查是否超過 15 個字元。

Error : ID-(0): the ID is too long , size be in under 15 words.

整串 ID 字數超過 15 個字元。

2. 一個識別字 的第一個字元必須是英文字母 a-zA-z 或是底線符號 _ 開始，在第一個字元之後，可以是英文字母、數字和底線符號。

識別字裡不可包含空白字元。

Rules : ({alpha}|_)+({alpha}|{digit}|_)*

篩選必須是以英文單字或底線符號開頭，在之後不管是不是英文或者是數字都可以，也有在 fuction 中做檢查。

例如：

peter 、 _db 、 a1 這些 是合法的識別字。

針對不合法的識別字，考慮 「數字」、「^」或「#」開頭，與長度過長這四種:

Rules : (\^|\#|{digit})+{id}

除了長度過長是在正確的 ID 中再去做判斷，其他的我是選擇在設一個另外的 Rules 去做篩選，而 lex 的規定中，會選擇較長的那個去

做 Patten，因此並不會跟 ID 做搞混。

Error : ID-(1) :開頭並非是英文字母 a-zA-z 或是底線符號(_)開始

Error : ID-(2) :開頭出現「^」或「#」的情況

例如:

1a 、 ^db 、 #db 、 abcdefghijklmnopqrstuvwxyz12345 這些都不是合法的識別字。

(3). 符號 (Symbols)

符號列表： : ( ) > < = == <= >= [+ *[

因為在其他測資中，會考慮到 1+2 等其他問題，因此單獨出現符號的情況，才會抓這個 token，所以我選擇將這個放在最後面，避免和其他的運算符號衝突。

(4). 整數 (Integer)

1.常數可以有正負，數字會緊接在正負號後，因此

Rules : Integer [+-]*{digit}+

如： 100, +20, 1000 。

2.針對「 1+2 」要判別為 整數、符號、整數，不能為「 1 」「 +2 」的情況，以特別的規定去抓到這個 token，在整數之前會緊臨一個整數:

Rules :　{Integer}[+-]{digit}+

2.針對在整數的定義，數字之前不會出現 0 的情況，因此我設定一個 rules:在 0 之後會緊臨一個整數，去抓到這個 token:

Rules :　[+-]*[0]+{digit}+

Error : Integer -(1) : You shouldnt set '0' in front of the integer with digits , it doesnt make sence.

(5). 實數 (Real)

常數可以有正負，且有小數點 (decimal point) 表示法和科學符號表示法兩種。

Rule:

real

[+-]?({small}|{small}[e|E][+-]{digit}+|{Integer}[e|E][+-]{digit}+)

我先用 small(小數)和(Ee)和 digit(數字)跟 Integer(整數)做結合，表示的方式也比較清楚:

小數: small　　　　{Integer}\.{digit}+

科學記號:E 或 e

例如：

1.0 、 3.14 、 7E-2 、 12.25e+6 、 -7.5E+3 這些是合法的實數

常數。

在小數 small 中:

1. 先找到『.』的位置，並且確認在這之前或之後是否包含 0。

2. real wrong :(1)

   如果一開始是 0 開頭，但在之後不是小數點(.)，為錯誤輸入。

例如：03.0

3. real wrong :(2-2)

   在小數點(.)之後，若最後末端連續出現兩個 00，為錯誤輸入。

例如：1.00 、12.100

4. real wrong :(3)

   沒有出現小數點(.)

5. real wrong :(4)

   如果一開始是(.)開頭，也直接為錯誤輸入。

6. real wrong :(5)

   Error : real wrong :(5) : You should put integer beforehand the dot(.) .

   需要在整數(小於 10^{-1})之前，也就是放置小數點(.)之前放置數字

   例如: .1 為錯誤輸入，不是合法的實數常數。

7. real wrong :(6)

Error : real wrong : (6) : You should put integer afterhand the dot(.) .

需要在整數(小於 10^{-1})之後，也就是放置小數點(.)之後放置數字

例如: 1. 為錯誤輸入，不是合法的實數常數。

(6). 字串常數 (quoted string)

Rule: \'([^\n]*)+\'，只要非換行都讀入，並且以’ ’ 為起始標誌和終止標誌。

1.字數最長到 30 個字元

Error : string-(1) : the string is too long , size be in under 30 words.

2.特殊問題能處理下列這些字串：

空字串，即 “ 或 ‘’ ，即一個空白字串。

3.針對這個問題:

‘You’ ’ ll see’ 代表字串 You’ ll see。

我使用 C 語言，先將完整的字串寫入，然後依靠判斷的方式，將字串做修整。

5. 單邊引號的不合法 srting

如「 ‘ab」和「ab’ 」需要判定為一個不合法的字串。

Rule 1 :\'[^\'\n\ \r]*

Error : string-(5-1) : You could forget to set (’)

afterhand the string what you tpye in .

Rule 2 : [^\’\n\ \r]*\’

Error : string-(5-2) : You could forget to set (’)

beforehand the string what you tpye in .

(7).**註解 Comment**

首先我將註解會需要用到的格式標示出來，以利我後續的作業。

Rule :

```
commentL       \(\*[*]*[^\*\)\n\r\ ]*
commentR       [^\n\r\(\*\ ]*[*]*\*\)
```

以上 2 個為不合法的註解定義，出現只有(*或*)的情況

```
commentLL      \(\*[*]*
commentRR      [*]*\*\)
```

以上 2 個為註解所需要的元素，為了下面的判斷所使用

comment

```
{commentLL}"("*([^*)]|([^*]")")|("*"[^)]?))*"*"*{commentRR}
```

當一個完整的註解，需要(*和*)同時出現並且符合左右對稱的定

義，不合法的情形就是兩者缺一(基數個的情況)。

2 種情況，例如:

1.(* comment *) 是一個合法的註解。沒有跨行，(*和*)各為一個

2.(* comment

second line *) 是一個合法的跨行註解。

特殊情況

1.(*****) 是一個合法的註解。

2.(* a**b) *) 是一個合法的註解。

因為符合註解定義，有偶數個合法的註解邊界(*或*)。

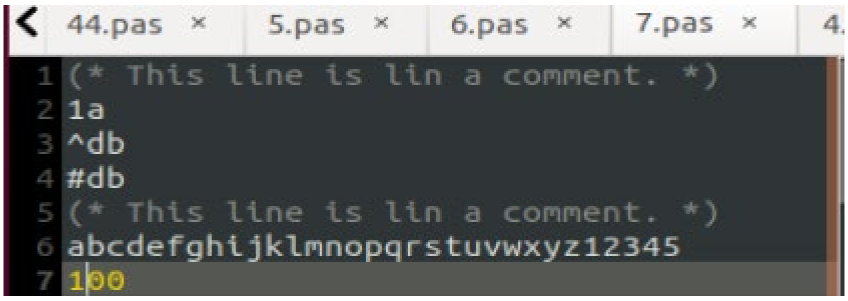
3.(*ab*)**) 不是一個合法的註解。

因為緊鄰出現基數個註解邊界(*或*)。

5. 遇到的問題

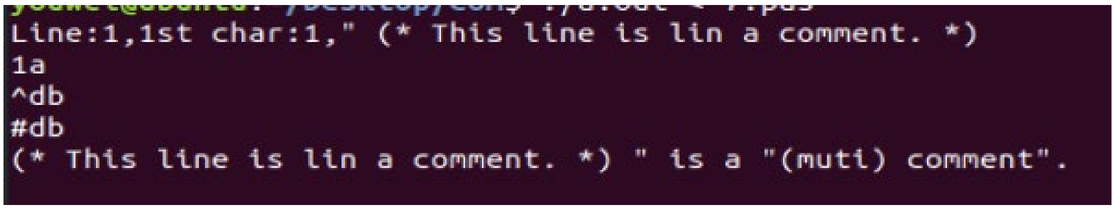
問題 1.

有 2 行分別不同行的 comment，要如何判斷成是分開的 comment



```
(* This line is lin a comment. *)
1a
^db
#db
(* This line is lin a comment. *)
abcdefghijklmnopqrstuvwxyz12345
100
```

會一次全部抓完



```
Line:1,1st char:1," (* This line is lin a comment. *)
1a
^db
#db
(* This line is lin a comment. *) " is a "(muti) comment".
```

問題 2.

一開始，遇到這個問題，很棘手，如果用上述的方式的話，會整個吃進 token，想問的是應該不是要我在 integer 這個 token 裡面的 rules 去標示 symbol，所以才想問這樣方法應該是有錯的

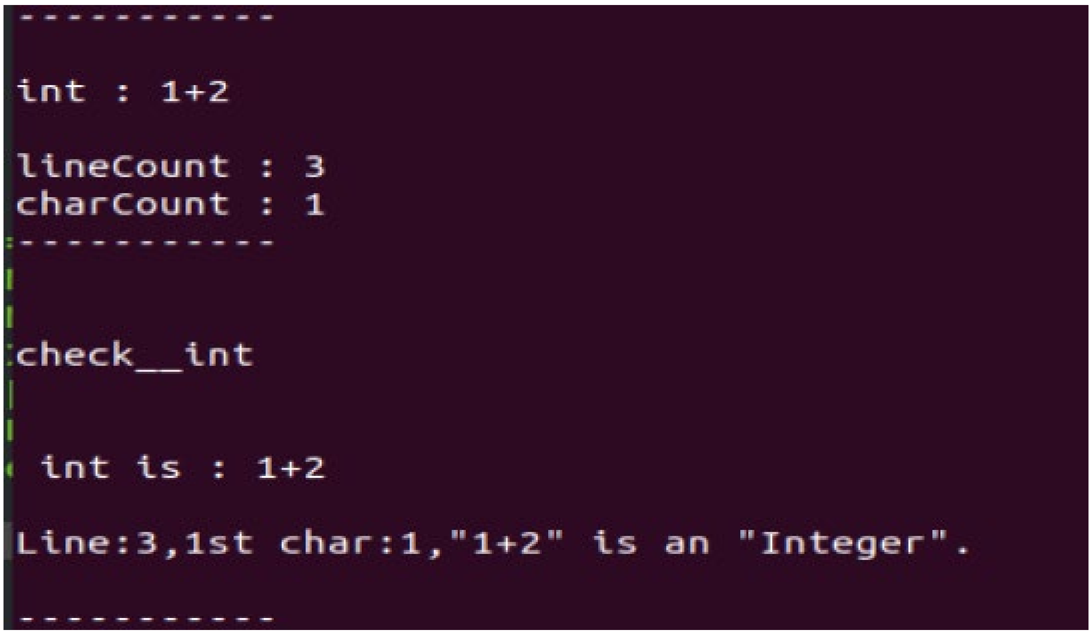
想過很多方法，想要在 C fuction 裡面直接解決，但是想到這門課是在教語法，所以想盡辦法要在 token 時候就先處理。

1+2

12+34 判斷會有錯誤

Integer   ([\+-]|[0-9])*[0-9]

解決方式:經過助教的提示，可以先利用較常和較嚴謹的表示法將所想要表達的 token 拿掉，就不會被之後的 token 吃掉



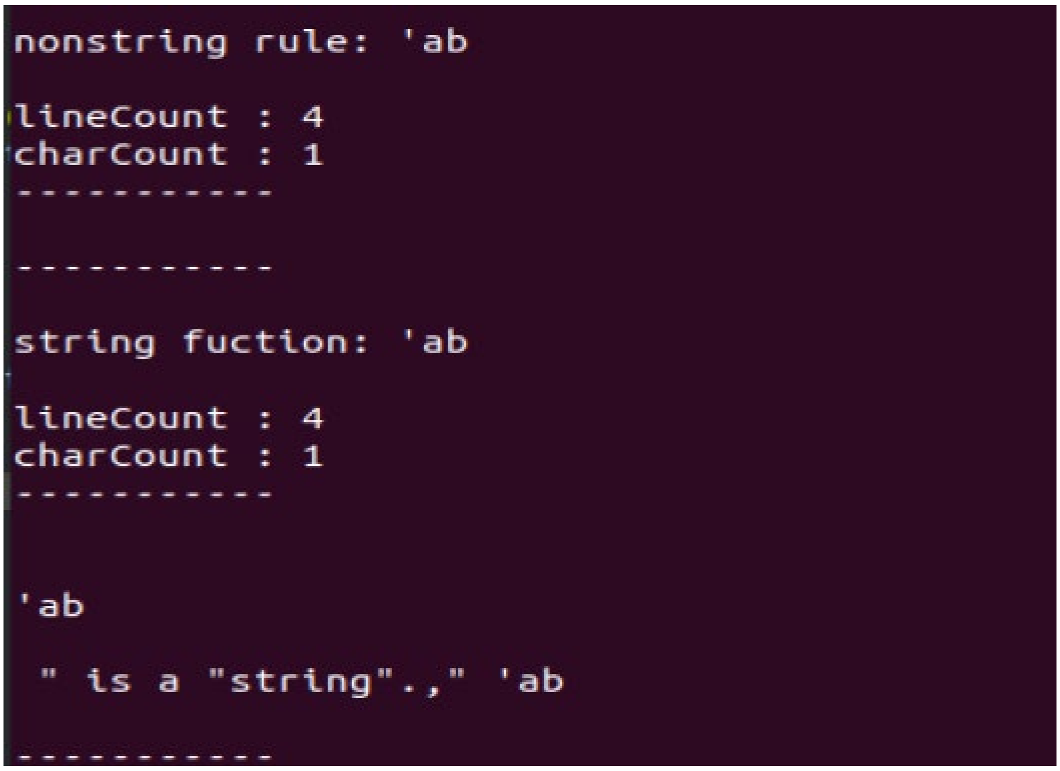
```
-----------

int : 1+2

lineCount : 3
charCount : 1
-----------


check__int


 int is : 1+2

Line:3,1st char:1,"1+2" is an "Integer".

-----------
```

問題 3

另外，這個條件下，’ ab 有吃到 nonstring 中，也有傳正確的行數和字數進去

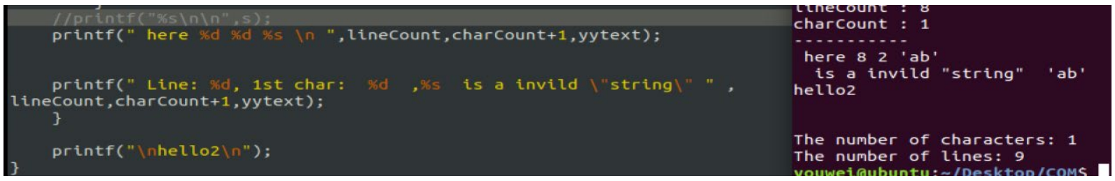
但是在 C fuction 中，沒有辦法顯示

‘ab



```
nonstring rule: 'ab

lineCount : 4
charCount : 1
-----------

-----------

string fuction: 'ab

lineCount : 4
charCount : 1
-----------


'ab

 " is a "string".," 'ab

-----------
```

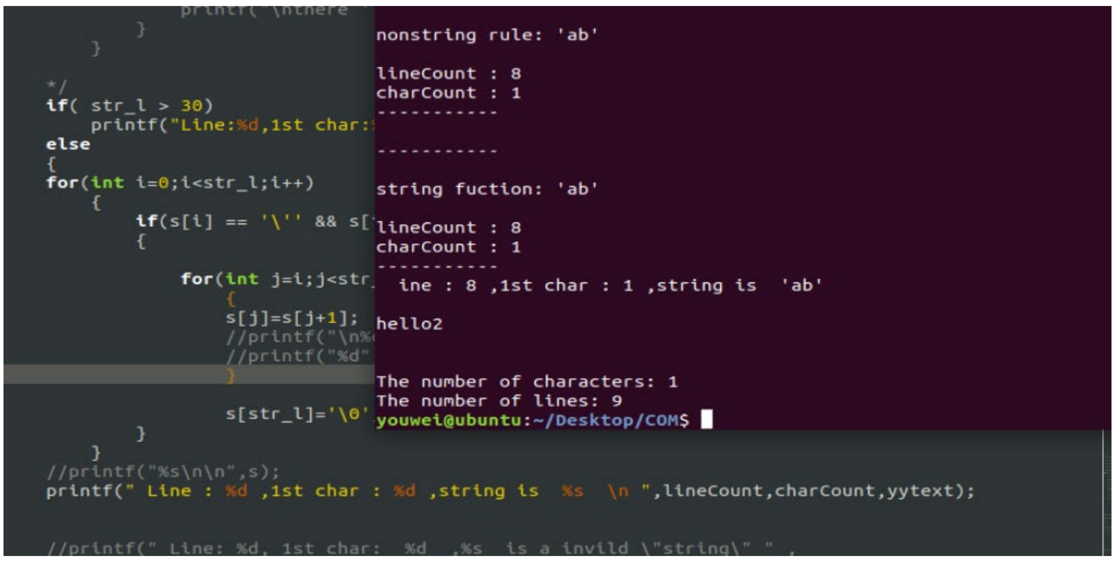
可是我上面放同樣的東西去檢查，結果是正確的，請問是哪裡出錯了呢?



```
    //printf("%s\n\n",s);
    printf(" here %d %d %s \n ",lineCount,charCount+1,yytext);


    printf(" Line: %d, 1st char:  %d  ,%s  is a invild \"string\" " ,
lineCount,charCount+1,yytext);
    }

    printf("\nhello2\n");
}
```

```
lineCount : 8
charCount : 1
-----------
 here 8 2 'ab'
  is a invild "string"  'ab'
hello2


The number of characters: 1
The number of lines: 9
youwei@ubuntu:~/Desktop/COM$
```

反覆測試幾次還是會吃自

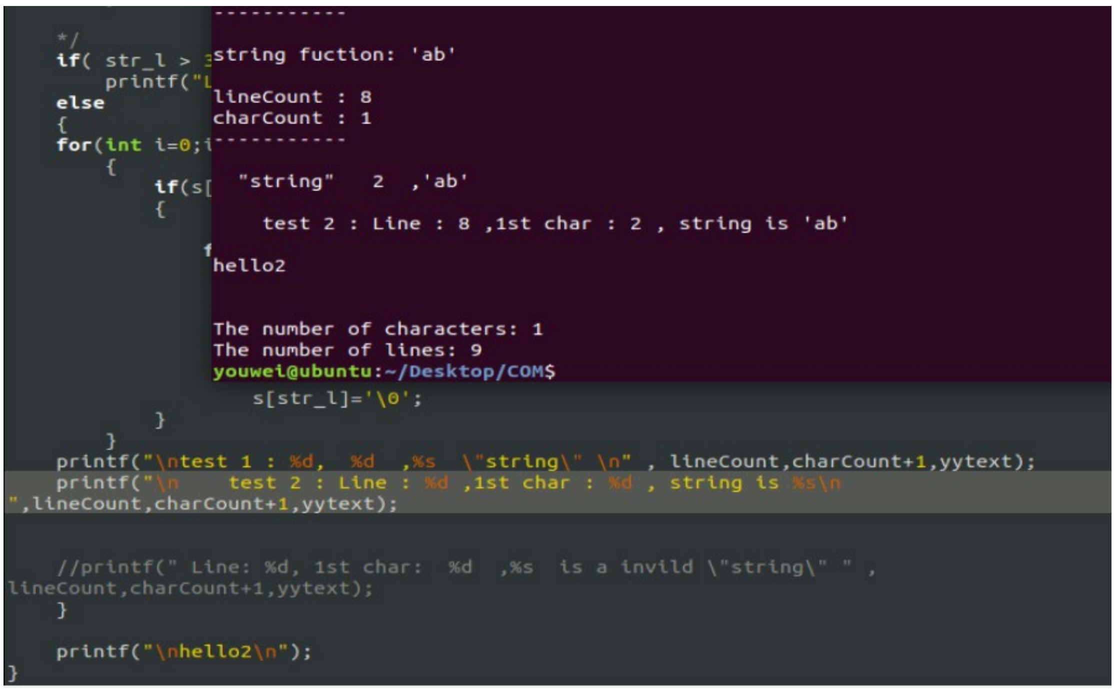


```
        }
    }

    */
    if( str_l > 30)
        printf("Line:%d,1st char:
    else
    {
    for(int i=0;i<str_l;i++)
        {
            if(s[i] == '\'' && s[
            {

                for(int j=i;j<str_
                    {
                    s[j]=s[j+1];
                    //printf("\n%
                    //printf("%d"
                    }

                    s[str_l]='\0'
            }
        }
    //printf("%s\n\n",s);
    printf(" Line : %d ,1st char : %d ,string is  %s  \n ",lineCount,charCount,yytext);

    //printf(" Line: %d, 1st char:  %d  ,%s  is a invild \"string\" " ,
```

```
nonstring rule: 'ab'

lineCount : 8
charCount : 1
-----------

-----------

string fuction: 'ab'

lineCount : 8
charCount : 1
-----------
 ine : 8 ,1st char : 1 ,string is  'ab'

hello2


The number of characters: 1
The number of lines: 9
youwei@ubuntu:~/Desktop/COM$
```

測試結果:

1.

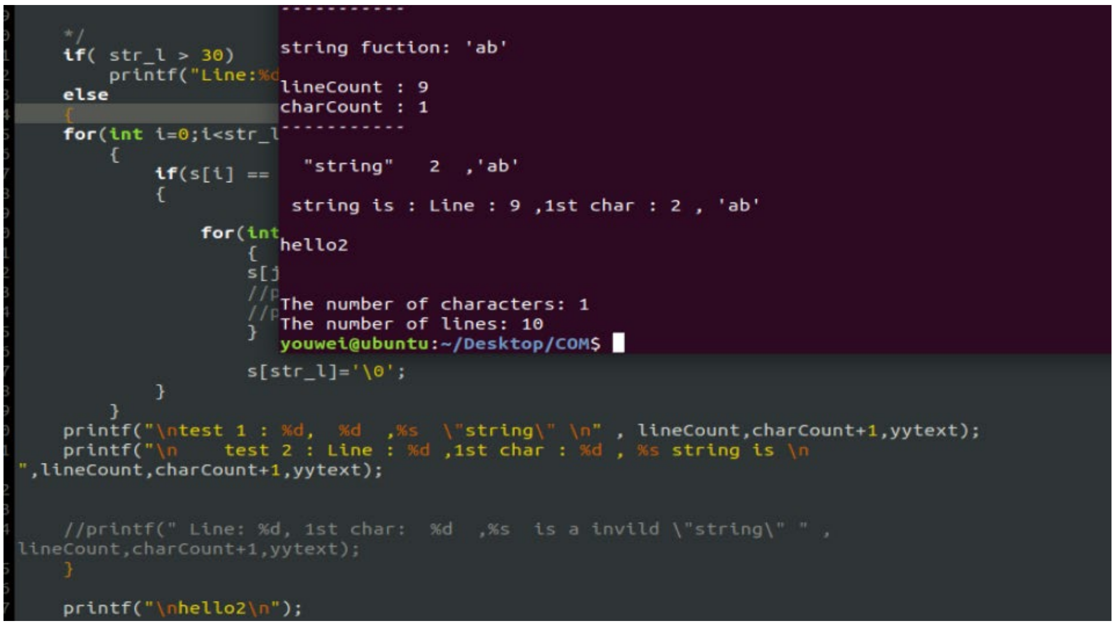


```
    */
    if( str_l > 3
        printf("L
    else
    {
    for(int i=0;i
        {
            if(s[
            {

                f
                    s[str_l]='\0';
            }
        }
    printf("\ntest 1 : %d,  %d  ,%s  \"string\" \n" , lineCount,charCount+1,yytext);
    printf("\n    test 2 : Line : %d ,1st char : %d , string is %s\n
",lineCount,charCount+1,yytext);


    //printf(" Line: %d, 1st char:  %d  ,%s  is a invild \"string\" " ,
lineCount,charCount+1,yytext);
    }

    printf("\nhello2\n");
}
```

```
-----------

string fuction: 'ab'

lineCount : 8
charCount : 1
-----------

 "string"   2  ,'ab'

   test 2 : Line : 8 ,1st char : 2 , string is 'ab'

hello2


The number of characters: 1
The number of lines: 9
youwei@ubuntu:~/Desktop/COM$
```

用這個方法，test2 前面要多放空白 不然會被吃掉

2.但是如果我將，string is 放在後面 又會被前面的吃掉



```
-----------

string fuction: 'ab'

lineCount : 9
charCount : 1
-----------

  "string"   2  ,'ab'

 string is : Line : 9 ,1st char : 2 , 'ab'

hello2


The number of characters: 1
The number of lines: 10
youwei@ubuntu:~/Desktop/COM$
```

```
    */
    if( str_l > 30)
        printf("Line:%d
    else
    {
    for(int i=0;i<str_l
        {
            if(s[i] ==
            {

                for(int
                    {
                    s[j
                    //p
                    //p
                    }
                    s[str_l]='\0';
            }
        }
    printf("\ntest 1 : %d,  %d  ,%s  \"string\" \n" , lineCount,charCount+1,yytext);
    printf("\n    test 2 : Line : %d ,1st char : %d , %s string is \n
",lineCount,charCount+1,yytext);


    //printf(" Line: %d, 1st char:  %d  ,%s  is a invild \"string\" " ,
lineCount,charCount+1,yytext);
    }

    printf("\nhello2\n");
```

解決方式:原來是因為\r 會將該行最前面的東西覆蓋過去，後面的

token 也會吃掉(yytext)

//alpha 　　　　　[A-Za-z] just for elemnts don't need

rules

問題 4

上顏色，這個應該算是加分項?

一開始，使用 color 函數，但是沒有辦法正確輸出效果，且部分輸

出只支援 windows 輸出，因此只好選擇用:

用 VT 碼，例：printf("\033[40;31m 你要改變顏色的內容

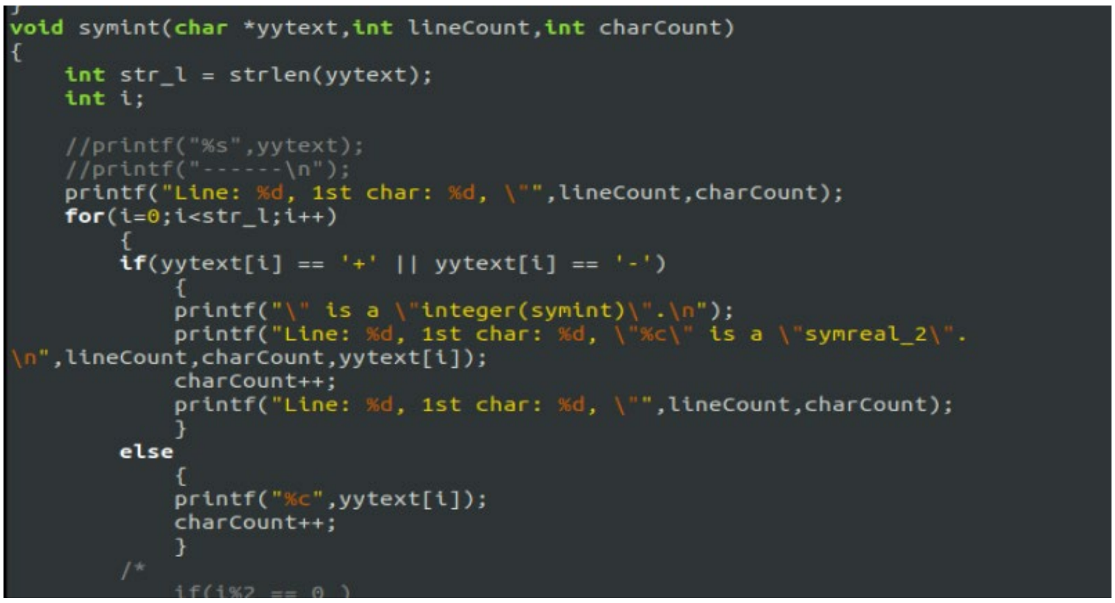
\033[0m"); 背景色為黑色，字體顏色位元紅色

只想要背景色 printf("\033[40m 你要改變顏色的內容\033[0m");

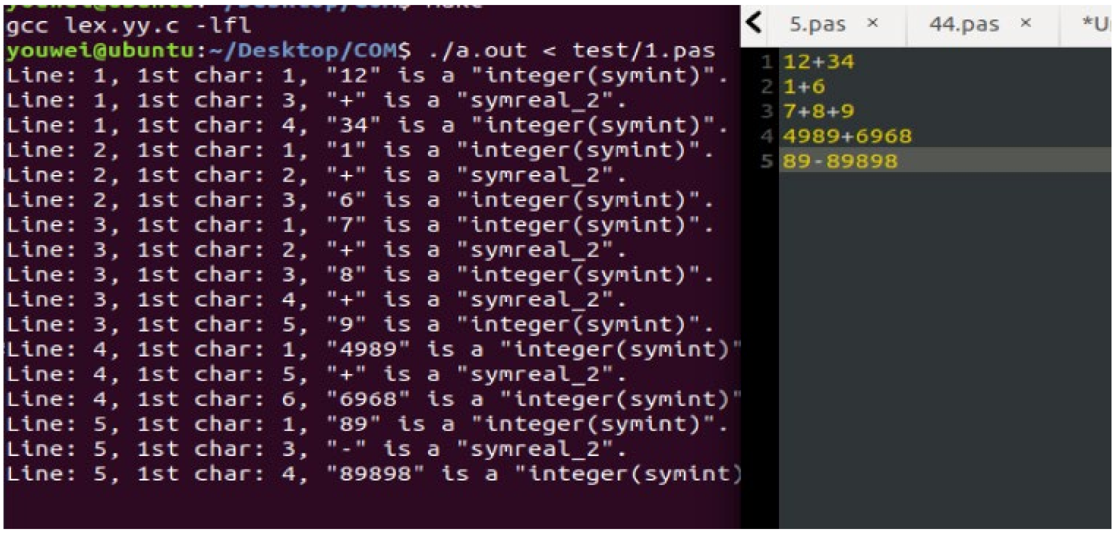
只想要自提顏色同上 改一下數字就行了

問題 5

還會有\r 吃字的情況



```
void symint(char *yytext,int lineCount,int charCount)
{
    int str_l = strlen(yytext);
    int i;

    //printf("%s",yytext);
    //printf("------\n");
    printf("Line: %d, 1st char: %d, \"",lineCount,charCount);
    for(i=0;i<str_l;i++)
        {
        if(yytext[i] == '+' || yytext[i] == '-')
            {
            printf("\" is a \"integer(symint)\".\n");
            printf("Line: %d, 1st char: %d, \"%c\" is a \"symreal_2\".
\n",lineCount,charCount,yytext[i]);
            charCount++;
            printf("Line: %d, 1st char: %d, \"",lineCount,charCount);
            }
        else
            {
            printf("%c",yytext[i]);
            charCount++;
            }
        /*
            if(i%2 == 0 )
```

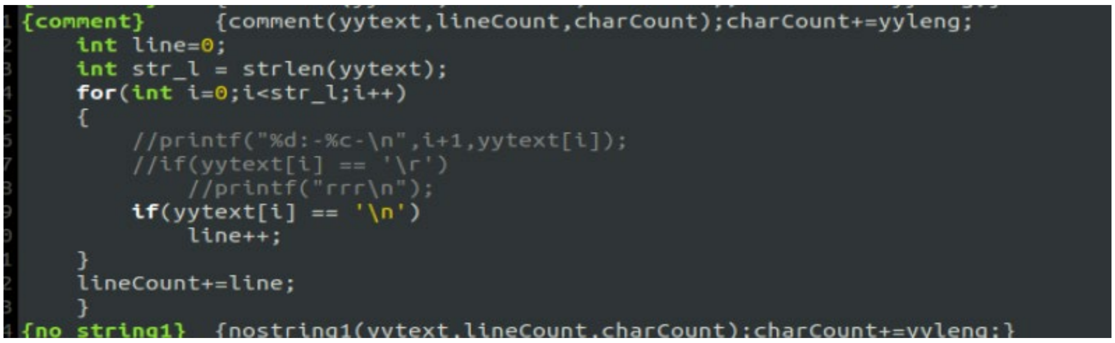


```
gcc lex.yy.c -lfl
youwei@ubuntu:~/Desktop/COM$ ./a.out < test/1.pas
Line: 1, 1st char: 1, "12" is a "integer(symint)".
Line: 1, 1st char: 3, "+" is a "symreal_2".
Line: 1, 1st char: 4, "34" is a "integer(symint)".
Line: 2, 1st char: 1, "1" is a "integer(symint)".
Line: 2, 1st char: 2, "+" is a "symreal_2".
Line: 2, 1st char: 3, "6" is a "integer(symint)".
Line: 3, 1st char: 1, "7" is a "integer(symint)".
Line: 3, 1st char: 2, "+" is a "symreal_2".
Line: 3, 1st char: 3, "8" is a "integer(symint)".
Line: 3, 1st char: 4, "+" is a "symreal_2".
Line: 3, 1st char: 5, "9" is a "integer(symint)".
Line: 4, 1st char: 1, "4989" is a "integer(symint)"
Line: 4, 1st char: 5, "+" is a "symreal_2".
Line: 4, 1st char: 6, "6968" is a "integer(symint)"
Line: 5, 1st char: 1, "89" is a "integer(symint)".
Line: 5, 1st char: 3, "-" is a "symreal_2".
Line: 5, 1st char: 4, "89898" is a "integer(symint)
```

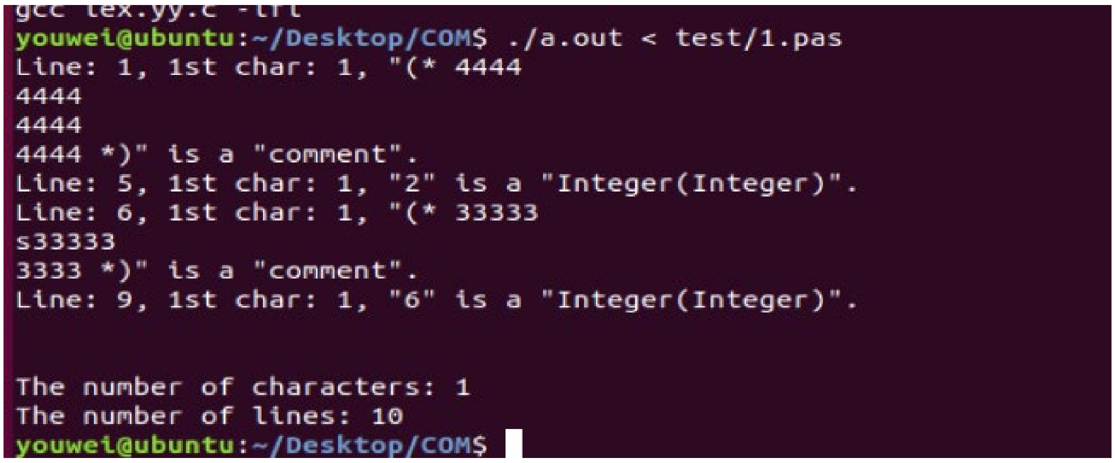
```
12+34
1+6
7+8+9
4989+6968
89-89898
```

問題 6.

多行的註解 Comment 換行會沒有計算到行數，而且在計算過程會受到\r 的影響導致輸出結果被覆蓋過去，學到教訓是 token 處理的時候\n\r 都要一起處理，不然會遇到很多消失字元的情況。



```
{comment}     {comment(yytext,lineCount,charCount);charCount+=yyleng;
    int line=0;
    int str_l = strlen(yytext);
    for(int i=0;i<str_l;i++)
    {
        //printf("%d:-%c-\n",i+1,yytext[i]);
        //if(yytext[i] == '\r')
            //printf("rrr\n");
        if(yytext[i] == '\n')
            line++;
    }
    lineCount+=line;
    }
{no_string1}  {nostring1(yytext,lineCount,charCount);charCount+=yyleng;}
```



```
gcc lex.yy.c -lfl
youwei@ubuntu:~/Desktop/COM$ ./a.out < test/1.pas
Line: 1, 1st char: 1, "(* 4444
4444
4444
4444 *)" is a "comment".
Line: 5, 1st char: 1, "2" is a "Integer(Integer)".
Line: 6, 1st char: 1, "(* 33333
s33333
3333 *)" is a "comment".
Line: 9, 1st char: 1, "6" is a "Integer(Integer)".


The number of characters: 1
The number of lines: 10
youwei@ubuntu:~/Desktop/COM$ 
```

問題 7

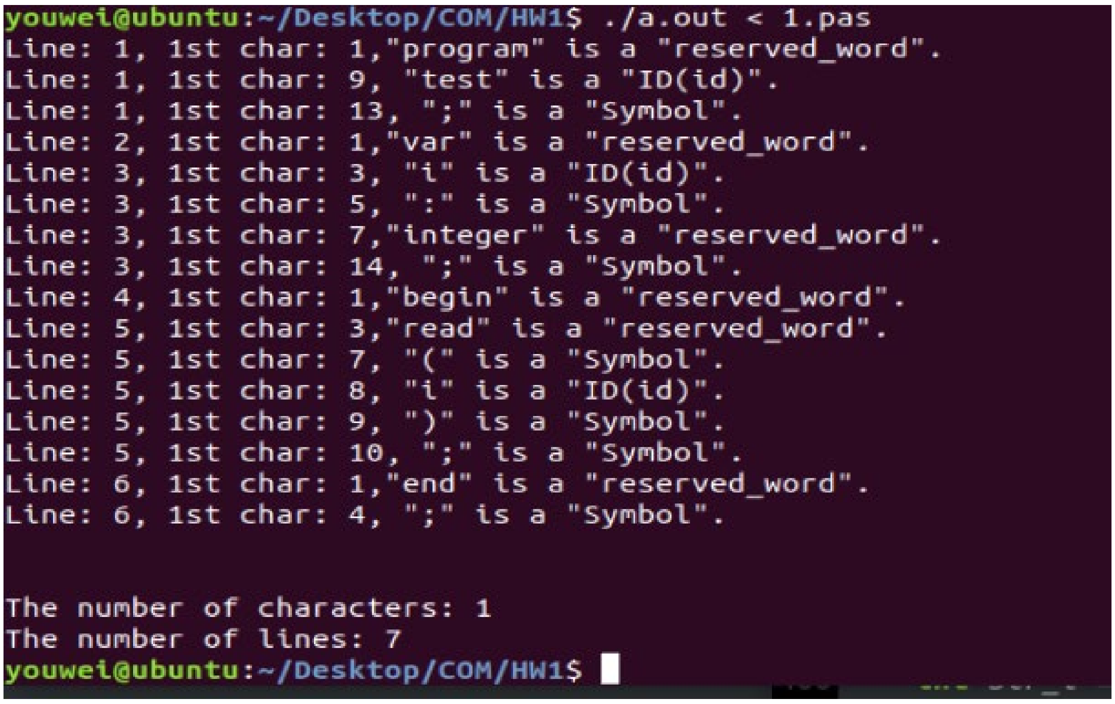
有寫 you’’ll 轉成 you’ll，這個應該算是加分項?

問題 8//Reserved_word C code

Reserved word 先用 C 語言，寫成程式，直接輸出結果，將來助教要再增加字，可以直接輸入，會轉成 lex 的表示式，大小寫。

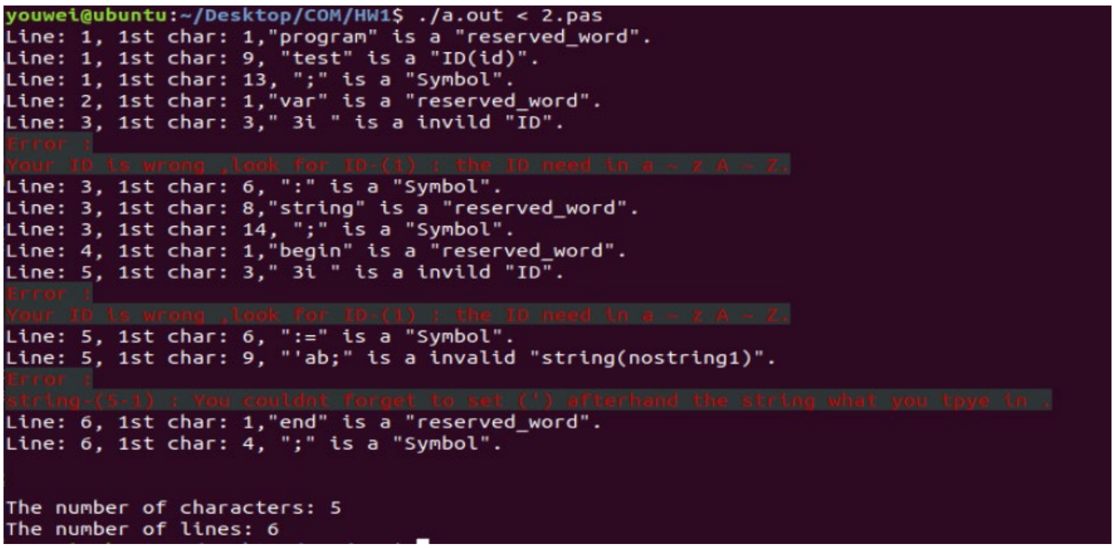
測試檔案結果:

1. Pas



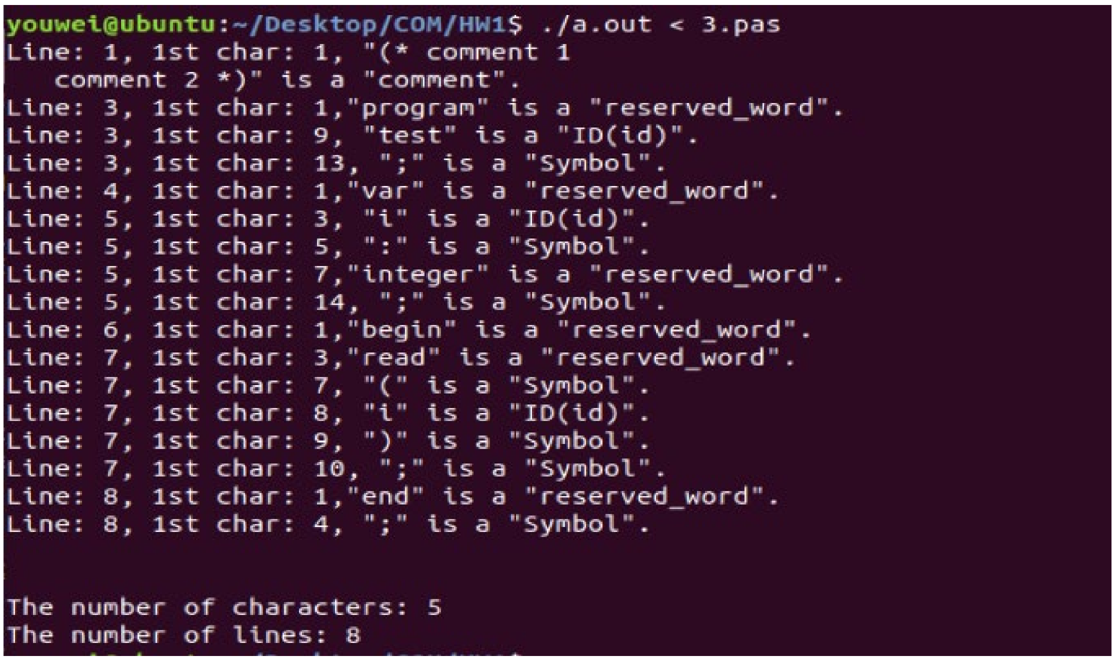
```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 1.pas
Line: 1, 1st char: 1,"program" is a "reserved_word".
Line: 1, 1st char: 9, "test" is a "ID(id)".
Line: 1, 1st char: 13, ";" is a "Symbol".
Line: 2, 1st char: 1,"var" is a "reserved_word".
Line: 3, 1st char: 3, "i" is a "ID(id)".
Line: 3, 1st char: 5, ":" is a "Symbol".
Line: 3, 1st char: 7,"integer" is a "reserved_word".
Line: 3, 1st char: 14, ";" is a "Symbol".
Line: 4, 1st char: 1,"begin" is a "reserved_word".
Line: 5, 1st char: 3,"read" is a "reserved_word".
Line: 5, 1st char: 7, "(" is a "Symbol".
Line: 5, 1st char: 8, "i" is a "ID(id)".
Line: 5, 1st char: 9, ")" is a "Symbol".
Line: 5, 1st char: 10, ";" is a "Symbol".
Line: 6, 1st char: 1,"end" is a "reserved_word".
Line: 6, 1st char: 4, ";" is a "Symbol".


The number of characters: 1
The number of lines: 7
youwei@ubuntu:~/Desktop/COM/HW1$
```

2. Pas



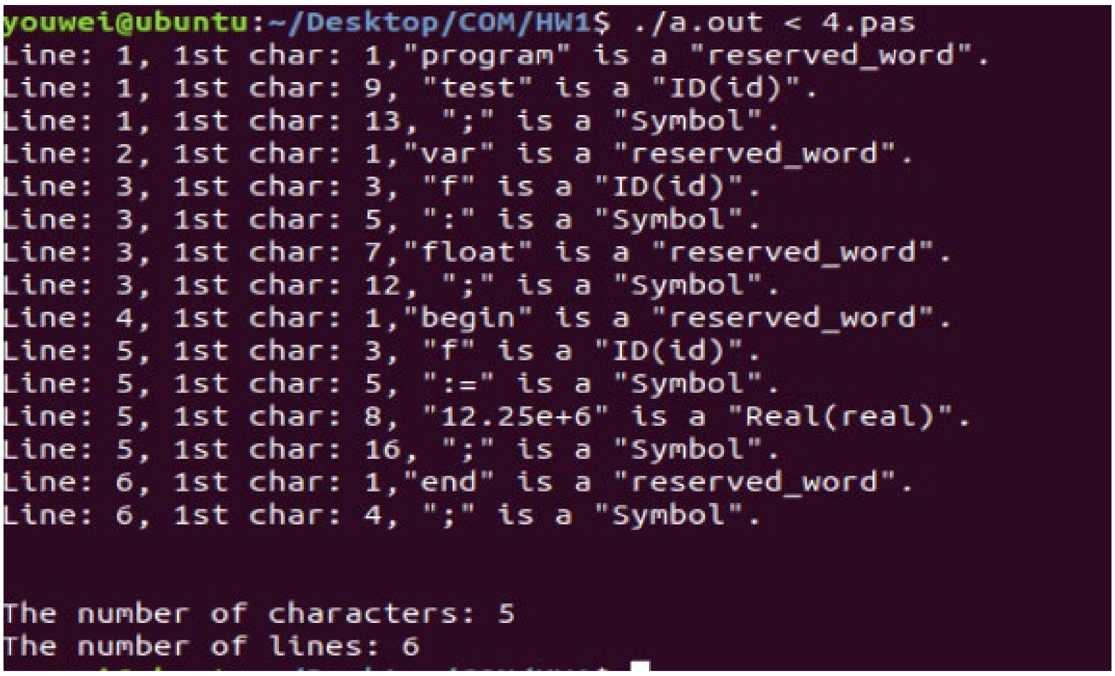
```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 2.pas
Line: 1, 1st char: 1,"program" is a "reserved_word".
Line: 1, 1st char: 9, "test" is a "ID(id)".
Line: 1, 1st char: 13, ";" is a "Symbol".
Line: 2, 1st char: 1,"var" is a "reserved_word".
Line: 3, 1st char: 3," 3i " is a invild "ID".
Error :
Your ID is wrong ,look for ID-(1) : the ID need in a ~ z A ~ Z.
Line: 3, 1st char: 6, ":" is a "Symbol".
Line: 3, 1st char: 8,"string" is a "reserved_word".
Line: 3, 1st char: 14, ";" is a "Symbol".
Line: 4, 1st char: 1,"begin" is a "reserved_word".
Line: 5, 1st char: 3," 3i " is a invild "ID".
Error :
Your ID is wrong ,look for ID-(1) : the ID need in a ~ z A ~ Z.
Line: 5, 1st char: 6, ":=" is a "Symbol".
Line: 5, 1st char: 9, "'ab;" is a invalid "string(nostring1)".
Error :
string-(5-1) : You couldnt forget to set (') afterhand the string what you tpye in .
Line: 6, 1st char: 1,"end" is a "reserved_word".
Line: 6, 1st char: 4, ";" is a "Symbol".


The number of characters: 5
The number of lines: 6
```

3. Pas



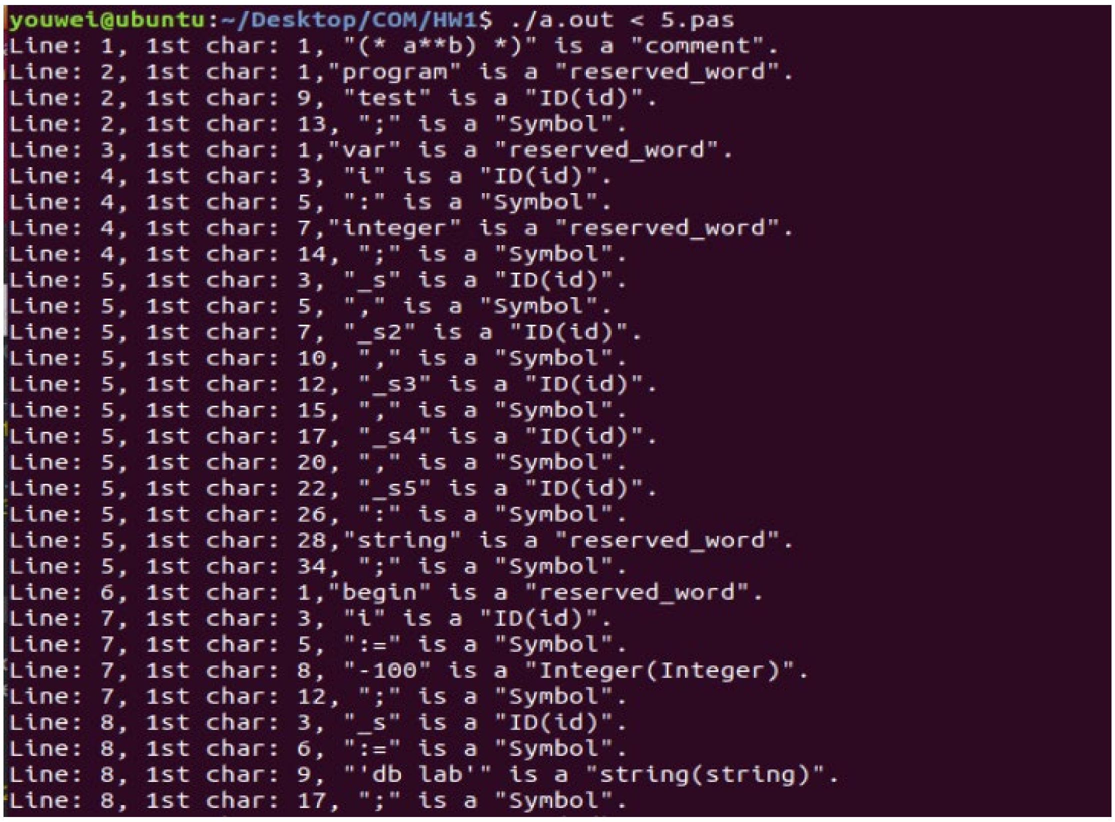
```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 3.pas
Line: 1, 1st char: 1, "(* comment 1
   comment 2 *)" is a "comment".
Line: 3, 1st char: 1,"program" is a "reserved_word".
Line: 3, 1st char: 9, "test" is a "ID(id)".
Line: 3, 1st char: 13, ";" is a "Symbol".
Line: 4, 1st char: 1,"var" is a "reserved_word".
Line: 5, 1st char: 3, "i" is a "ID(id)".
Line: 5, 1st char: 5, ":" is a "Symbol".
Line: 5, 1st char: 7,"integer" is a "reserved_word".
Line: 5, 1st char: 14, ";" is a "Symbol".
Line: 6, 1st char: 1,"begin" is a "reserved_word".
Line: 7, 1st char: 3,"read" is a "reserved_word".
Line: 7, 1st char: 7, "(" is a "Symbol".
Line: 7, 1st char: 8, "i" is a "ID(id)".
Line: 7, 1st char: 9, ")" is a "Symbol".
Line: 7, 1st char: 10, ";" is a "Symbol".
Line: 8, 1st char: 1,"end" is a "reserved_word".
Line: 8, 1st char: 4, ";" is a "Symbol".


The number of characters: 5
The number of lines: 8
```

4. Pas

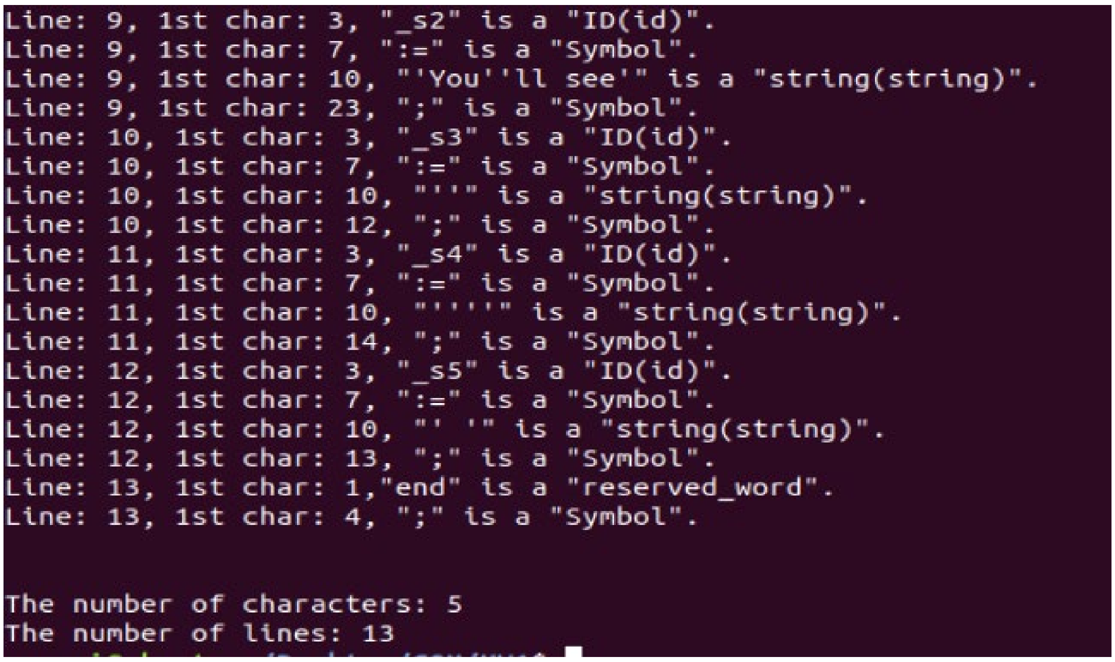


```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 4.pas
Line: 1, 1st char: 1,"program" is a "reserved_word".
Line: 1, 1st char: 9, "test" is a "ID(id)".
Line: 1, 1st char: 13, ";" is a "Symbol".
Line: 2, 1st char: 1,"var" is a "reserved_word".
Line: 3, 1st char: 3, "f" is a "ID(id)".
Line: 3, 1st char: 5, ":" is a "Symbol".
Line: 3, 1st char: 7,"float" is a "reserved_word".
Line: 3, 1st char: 12, ";" is a "Symbol".
Line: 4, 1st char: 1,"begin" is a "reserved_word".
Line: 5, 1st char: 3, "f" is a "ID(id)".
Line: 5, 1st char: 5, ":=" is a "Symbol".
Line: 5, 1st char: 8, "12.25e+6" is a "Real(real)".
Line: 5, 1st char: 16, ";" is a "Symbol".
Line: 6, 1st char: 1,"end" is a "reserved_word".
Line: 6, 1st char: 4, ";" is a "Symbol".


The number of characters: 5
The number of lines: 6
```

5. Pas

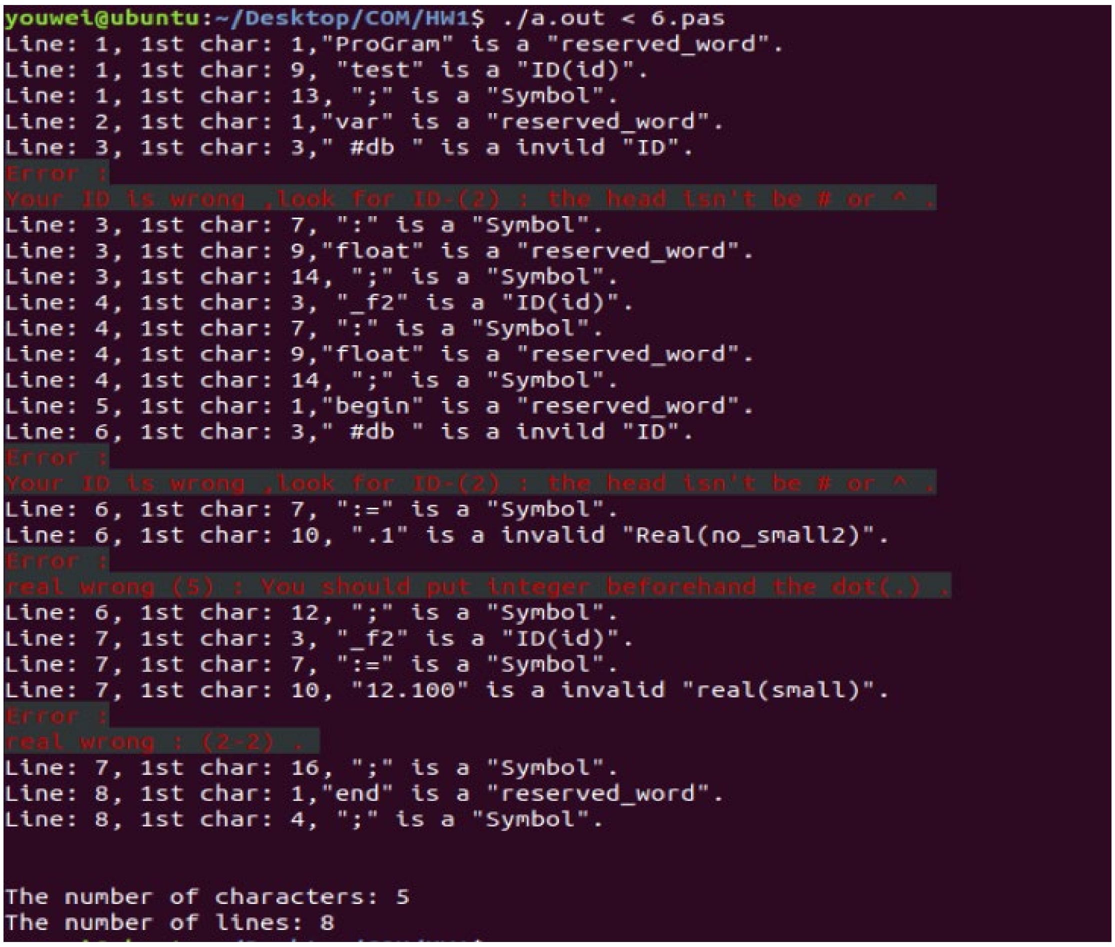


```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 5.pas
Line: 1, 1st char: 1, "(* a**b) *)" is a "comment".
Line: 2, 1st char: 1,"program" is a "reserved_word".
Line: 2, 1st char: 9, "test" is a "ID(id)".
Line: 2, 1st char: 13, ";" is a "Symbol".
Line: 3, 1st char: 1,"var" is a "reserved_word".
Line: 4, 1st char: 3, "i" is a "ID(id)".
Line: 4, 1st char: 5, ":" is a "Symbol".
Line: 4, 1st char: 7,"integer" is a "reserved_word".
Line: 4, 1st char: 14, ";" is a "Symbol".
Line: 5, 1st char: 3, "_s" is a "ID(id)".
Line: 5, 1st char: 5, "," is a "Symbol".
Line: 5, 1st char: 7, "_s2" is a "ID(id)".
Line: 5, 1st char: 10, "," is a "Symbol".
Line: 5, 1st char: 12, "_s3" is a "ID(id)".
Line: 5, 1st char: 15, "," is a "Symbol".
Line: 5, 1st char: 17, "_s4" is a "ID(id)".
Line: 5, 1st char: 20, "," is a "Symbol".
Line: 5, 1st char: 22, "_s5" is a "ID(id)".
Line: 5, 1st char: 26, ":" is a "Symbol".
Line: 5, 1st char: 28,"string" is a "reserved_word".
Line: 5, 1st char: 34, ";" is a "Symbol".
Line: 6, 1st char: 1,"begin" is a "reserved_word".
Line: 7, 1st char: 3, "i" is a "ID(id)".
Line: 7, 1st char: 5, ":=" is a "Symbol".
Line: 7, 1st char: 8, "-100" is a "Integer(Integer)".
Line: 7, 1st char: 12, ";" is a "Symbol".
Line: 8, 1st char: 3, "_s" is a "ID(id)".
Line: 8, 1st char: 6, ":=" is a "Symbol".
Line: 8, 1st char: 9, "'db lab'" is a "string(string)".
Line: 8, 1st char: 17, ";" is a "Symbol".
```



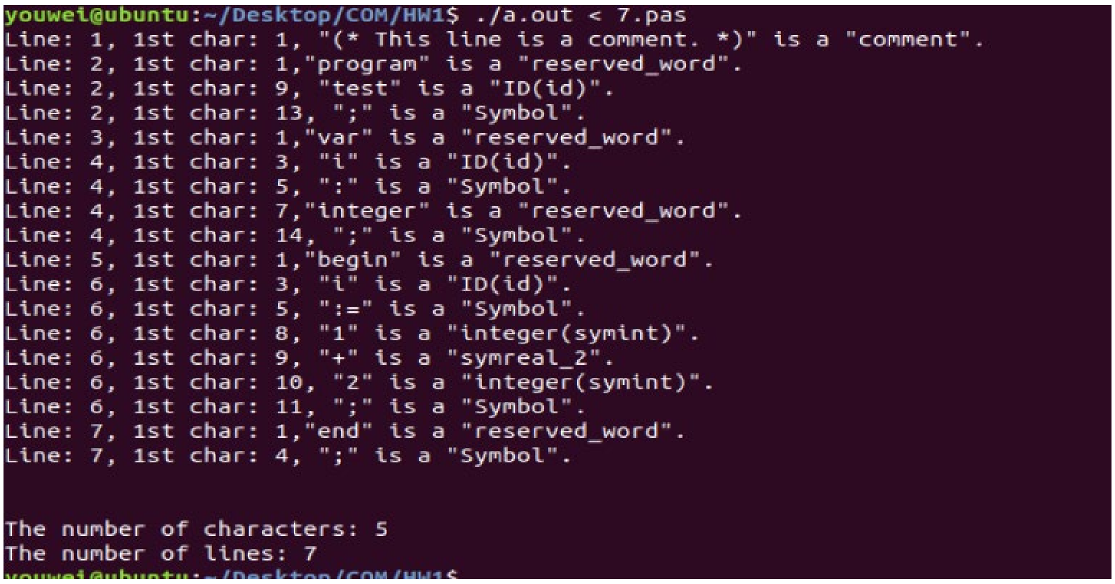
```
Line: 9, 1st char: 3, "_s2" is a "ID(id)".
Line: 9, 1st char: 7, ":=" is a "Symbol".
Line: 9, 1st char: 10, "'You''ll see'" is a "string(string)".
Line: 9, 1st char: 23, ";" is a "Symbol".
Line: 10, 1st char: 3, "_s3" is a "ID(id)".
Line: 10, 1st char: 7, ":=" is a "Symbol".
Line: 10, 1st char: 10, "''" is a "string(string)".
Line: 10, 1st char: 12, ";" is a "Symbol".
Line: 11, 1st char: 3, "_s4" is a "ID(id)".
Line: 11, 1st char: 7, ":=" is a "Symbol".
Line: 11, 1st char: 10, "''''" is a "string(string)".
Line: 11, 1st char: 14, ";" is a "Symbol".
Line: 12, 1st char: 3, "_s5" is a "ID(id)".
Line: 12, 1st char: 7, ":=" is a "Symbol".
Line: 12, 1st char: 10, "' '" is a "string(string)".
Line: 12, 1st char: 13, ";" is a "Symbol".
Line: 13, 1st char: 1,"end" is a "reserved_word".
Line: 13, 1st char: 4, ";" is a "Symbol".


The number of characters: 5
The number of lines: 13
```

6. Pas



```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 6.pas
Line: 1, 1st char: 1,"ProGram" is a "reserved_word".
Line: 1, 1st char: 9, "test" is a "ID(id)".
Line: 1, 1st char: 13, ";" is a "Symbol".
Line: 2, 1st char: 1,"var" is a "reserved_word".
Line: 3, 1st char: 3," #db " is a invild "ID".
Error :
Your ID is wrong ,look for ID-(2) : the head isn't be # or ^ .
Line: 3, 1st char: 7, ":" is a "Symbol".
Line: 3, 1st char: 9,"float" is a "reserved_word".
Line: 3, 1st char: 14, ";" is a "Symbol".
Line: 4, 1st char: 3, "_f2" is a "ID(id)".
Line: 4, 1st char: 7, ":" is a "Symbol".
Line: 4, 1st char: 9,"float" is a "reserved_word".
Line: 4, 1st char: 14, ";" is a "Symbol".
Line: 5, 1st char: 1,"begin" is a "reserved_word".
Line: 6, 1st char: 3," #db " is a invild "ID".
Error :
Your ID is wrong ,look for ID-(2) : the head isn't be # or ^ .
Line: 6, 1st char: 7, ":=" is a "Symbol".
Line: 6, 1st char: 10, ".1" is a invalid "Real(no_small2)".
Error :
real wrong (5) : You should put integer beforehand the dot(.) .
Line: 6, 1st char: 12, ";" is a "Symbol".
Line: 7, 1st char: 3, "_f2" is a "ID(id)".
Line: 7, 1st char: 7, ":=" is a "Symbol".
Line: 7, 1st char: 10, "12.100" is a invalid "real(small)".
Error :
real wrong : (2-2) .
Line: 7, 1st char: 16, ";" is a "Symbol".
Line: 8, 1st char: 1,"end" is a "reserved_word".
Line: 8, 1st char: 4, ";" is a "Symbol".


The number of characters: 5
The number of lines: 8
```

7. pas



```
youwei@ubuntu:~/Desktop/COM/HW1$ ./a.out < 7.pas
Line: 1, 1st char: 1, "(* This line is a comment. *)" is a "comment".
Line: 2, 1st char: 1,"program" is a "reserved_word".
Line: 2, 1st char: 9, "test" is a "ID(id)".
Line: 2, 1st char: 13, ";" is a "Symbol".
Line: 3, 1st char: 1,"var" is a "reserved_word".
Line: 4, 1st char: 3, "i" is a "ID(id)".
Line: 4, 1st char: 5, ":" is a "Symbol".
Line: 4, 1st char: 7,"integer" is a "reserved_word".
Line: 4, 1st char: 14, ";" is a "Symbol".
Line: 5, 1st char: 1,"begin" is a "reserved_word".
Line: 6, 1st char: 3, "i" is a "ID(id)".
Line: 6, 1st char: 5, ":=" is a "Symbol".
Line: 6, 1st char: 8, "1" is a "integer(symint)".
Line: 6, 1st char: 9, "+" is a "symreal_2".
Line: 6, 1st char: 10, "2" is a "integer(symint)".
Line: 6, 1st char: 11, ";" is a "Symbol".
Line: 7, 1st char: 1,"end" is a "reserved_word".
Line: 7, 1st char: 4, ";" is a "Symbol".


The number of characters: 5
The number of lines: 7
```