```
%{
   /* definition */
   #include <stdio.h>
   #include <ctype.h>

%}


%%
Start : N  L {printf("\n\n");}
      ;
N     : {$$ = 1;}
      ;
L : S {$$ = $1;}
  | L ',' M  S {$$ = $4;}
  ;
M : {$$ = ($-1) + 1;}
  ;
S : '(' K L ')' {$$ = $3 + 1;}
  | 'a' {printf(" a @%d \n", ($0)); $$ = ($0) +
  1;}
K : {$$ = ($-1) + 1;}
  ;
%%
/*
关于$0 以及 $-1 等YACC辅助符号：
如S的产生式中语义动作里出现$0，则其表示产生式左
部非终结符S"左边"的第一个符号X的属性，
也就是说，符号X，是S在别的产生式右部出现时的最近
```

左邻符号而已！！！
以此类推，若出现$-1，则表示S在别的产生式右部出现时的第2近左邻符号而已。

或者也可以理解为，$0是分析栈中左部非终结符位置下方的第一个符号，$-1，则是下方的第二个符号...

供参考的属性翻译方案

属性说明
L.p : 继承属性，L的第一个字符开始位置
L.s : 综合属性，L之后的字符开始位置

S.p : 继承属性，S的第一个字符开始位置
S.s : 综合属性，S之后的字符开始位置

N/M/L为空产生式，其属性：

N.s: 整个输入串的第一个字符位置，值为 1
M.p :
M.S :
K.p :
K.s :

这些属性的作用是将“字符位置”传递到紧邻L或S的句柄下方，这样一来，
当字符a被移入栈顶时（句柄），可固定地从该句柄下方所在读取“字符位置”的值并打印输出

```
Start : N {L.p = N.s} L

N     : {N.s = 1}

L     : {S.p = L.p;}
        S
        { L.s = S.s}

L     : {L1.p = L.p;}
        L1
        ','
        {M.p = L1.s;}
        M
        {S.p = M.s;}
        S
        {L.s = S.s;}

M     : {M.s = M.p + 1;}

S     :  '('
         {K.p = S.p;}
         K
         {L.p = K.s;}
         L
         ')'
         {S.s = L.s + 1;}
```

```
S     :  'a'
         {
           printf(" a @%d\n", S.p);
           S.s = S.p + 1;
         }

K     : {K.s = K.p + 1;}

*/
int yylex()
{
   int c;
   while ( ( c=getchar() ) == ' ' ) ;
   if ( c=='\n' ) return 0; // return EOF -- $
   return c;
}

int main()
{
   return yyparse();
}

void yyerror(char *s)
{
   printf("%s\n", s );
}

int yywrap()
{ return 1; }
```