%{ int l = 1;

%} %%

\n l++;

, printf("%d tCOMMA\n", l);

\"==\" printf("%d tEQUALITY\n", l);

\"\+\" printf("%d tPLUS\n", l);

\"\-\" printf("%d tMINUS\n", l);

\"\\*\" printf("%d tMUL\n", l);

\"\/\" printf("%d tDIV\n", l);

\"\+\+\" printf("%d tINC\n", l); \"\-\-\" printf("%d tDEC\n", l); \[ printf("%d tLBRAC\n", l);

\] printf("%d tRBRAC\n", l); \">\" printf("%d tGT\n", l);

\"<\" printf("%d tLT\n", l);

\">=\" printf("%d tGEQ\n", l); \"<=\" printf("%d tLEQ\n", l);

\"Get\" printf("%d tGET\n", l);

\"Set\" printf("%d tSET\n", l); \"Function\" printf("%d tFUNCTION\n", l); \"Print\" printf("%d tPRINT\n", l);

\"For\" printf("%d tFOR\n", l); \"If\" printf("%d tIF\n", l); \"Return\" printf("%d tRETURN\n, l"); \"[a-zA-Z\_][a-zA-Z0-9\_]\*\" { printf("%d tIDENT ", l); char \*p = yytext;

p++;

p[yyleng-2] = '\0';

printf("(%s)\n", p);

}

\'[^']\*\' printf("%d tSTRING (%s)\n", l, yytext);

-?[0-9]+\.[0-9]+ {

printf("%d tNUM (%s) ", l, yytext);

char \*p = yytext;

int i, j;

for(i = 0; i < yyleng; i++){

if(yytext[i] == '0' && yytext[i+1] == '0'){

p++;}

else{

break;}

}

for(j = yyleng-1; j > 0; j++){

if(yytext[j] == '0'){

p[j-1] = '\0';}

else{

break;}

}

printf("(%s)\n", p);}

-?[0-9]+ printf("%d tNUM (%s) (%d)\n", l, yytext, atoll(yytext));

.

%%