# Auxiliary functions

* Lex generates C code for the rules specified in the Rules section andplaces this code into a single function called yylex(). (To be discussed in detail later). In addition to this Lex generated code, the programmer may wish to add his own code to the lex.yy.c file. The auxiliary functions section allows the programmer to achieve this.

Example:

/\* Declarations \*/

%%

/\* Rules \*/

%%

int main()

{

yylex(); return 1;

}

# The yyvariables

The following variables are offered by Lex to aid the programmer in designing sophisticated lexical analyzers. These variables are accessible in the Lex program and are automatically declared by Lex in lex.yy.c.

1. yyin is a variable of the type FILE\* and points to the input file. yyin is defined by Lex automatically. If the programmer assigns an input file to yyin in the auxiliary functions section, then yyin is set to point to that file. Otherwise Lex assigns yyin to stdin(console input).
2. yytext is of type char\* and it contains the lexeme currently found. A lexeme is a sequence of characters in the input stream that matches some pattern in the Rules Section. (In fact, it is the first matching sequence in the input from the position pointed to by yyin.) Each invocation of the function yylex() results in yytext carrying a pointer to the lexeme found in the input stream by yylex(). The value of yytext will be overwritten after the next yylex() invocation.
3. yyleng is a variable of the type int and it stores the length of the lexeme pointed to by yytext.

# The yyfunctions

1. yylex()
   * yylex() is a function of return type int. Lex automatically defines yylex() in lex.yy.c but does not call it. The programmer must call yylex() in the Auxiliary functions section of the Lex program. Lex generates code for the definition of yylex() according to the rules specified in the Rules section.
   * yylex() need not necessarily be invoked in the Auxiliary Functions Section of Lex program when used with YACC.
2. yywrap()
   * Lex declares the function yywrap() of return-type int in the file lex.yy.c. Lex does not provide any definition for yywrap(). yylex() makes a call to yywrap() when it encounters the end of input.
   * If yywrap() returns zero (indicating false) yylex() assumes there is more input and it continues scanning from the location pointed to by yyin. If yywrap() returns a non-zero value (indicating true), yylex() terminates the scanning process and returns 0.

# Example of Lex Tool

%{

#include<stdio.h>

%}

%%

"hi" {printf("bye");}

.\* {printf("Wrong Input");}

%%

void main()

{ printf("Enter IP"); yylex();

}

**OUTPUT:**

