Exp No: 5.1.

STUDY OF LEX

AIM :

To study the working of LEX.

INTORDUCTION :

In lexical analysis the stream of characters making up the source code is read from left to right and grouped in to tokens that are sequence of characters having a collective meaning. Lex is a tool that has been widely used to specify lexical analyzer for a variety of languages.The tool is referred to as lex compiler and its input specification is referred to as the lex language. Lex is generally used in the following manner.First a specification of a lexical analyser is prepared by creating a program *lex.l* in the lex language . Then *lex.l* is run through the lex language to produce a c program *lex.yy.c*. The program lex.yy.cc consist of a tabular representation of a transition diagram constructed from the regular expression of *lex.l* together with a standard routine to that uses the table to recognize the lexemes. Finally *lex.yy.c* is run through the c compiler to produce an object program *a.out*, which is the lexical analyser that transform an input stream in to sequence of tokens.

Regular expressions (Examples)
. Match any character except newlines.
\n A newline character

\t A tab character.
^ The beginning of the line.
$ The end of the line.
<expr>* Zero or more occurences of the expression.
<expr>+ One or more occurences of the expression.
(<expr1>|<expr2>) One expression of another.
[<set>] A set of character or ranges, such as [a-zA-Z]

Structure of a lex file

A lex file looks like
declaration
%%
transition rules
%%
Auxiliary procedures

Declaration:

There are three things that can go in the declaration section:

- C code

Any indented code between
%{
and
%}
is copied to the C file. This is typically used for defining file variables, and for prototypes of routines that are defined in the code segment.

- Definitions

A definition is very much like a #define directive. For example
letter [a-zA-Z]
digit [0-9]
punct [,.:;!?]
nonblank [^ \t]
These definitions can be used in the rules section

Transition Rules section
The rules section has a number of pattern-action pairs. The patterns are regular expressions and the actions are either a single C command, or a sequence enclosed in braces.
If more than one rule matches the input, the longer match is taken. If two matches are the same length, the earlier one in the list is taken

Auxiliary procedures/User code section

If the lex program is to be used on its own, this section will contain a
main program. If you leave this section empty you will get the default main:

```
int main()
{
yylex();
return 0;
}
```

where yylex is the parser that is built from the rules

RESULT:
Studied the working of LEX .