LPCC Assignment 3

Name: Digvijay Pawar

Class: TY.Btech Comp B2

Gr.No: 21810344

Roll No: 322043

**Aim**: Write a program to implement a lexical analyzer for parts of speech using LEX.

### **Objective**: To identify parts of speech in a sentence.

**Theory**:

What is LexicaL ANALYSIS? Lexical analysis is the very first phase in the compiler designing. It takes the modified source code which is written in the form of sentences. In other words, it helps you to converts a sequence of characters into a sequence of tokens. The lexical analysis breaks this syntax into a series of tokens. It removes any extra space or comment written in the source code.

If the lexical analyser finds a token invalid, it generates an error. The lexical analyser works closely with the syntax analyser. It reads character streams from the source code, checks for legal tokens, and passes the data to the syntax analyser when it demands.  
 In programming language, keywords, constants, identifiers, strings, numbers, operators and punctuations symbols can be considered as tokens.

Code:

%{

%}

%%

Jay|Viru|John|Abdul|are

{printf("%s is a Noun",yytext);}

ran|walk|playing|dancing|sleeping

{printf("%s: is a verb\n",yytext);}

at|in|on|from|above|below

{printf("%s: is Preposition\n",yytext);}

effortlessly|simply|gently|quickly|slowly

{printf("%s: is a adverb\n",yytext);}

and|or|also|so|but|if|then

{printf("%s: is Conjuction\n",yytext);}

this|that|he|she|it|we|they|him|her|they

{printf("%s: is Pronoun\n",yytext);}

[a-zA-Z]+ {printf("%s: don't recognize\n",yytext);}

%%

int yywrap(){

return 1;

}

int main()

{

yylex();

yywrap();

return 0;

}

Output:

