**Name:Kundan Kabra  
Gr No: 21810729  
Roll No: 321023  
Div: A-1**

**CODE :-**

**3A :-**

%{

/\*

\*To implement a lexical analyzer for parts of speech without Symbol Table

\*/

%}

VERB is|am|are|brought|disappeared|likes|exhausted|touch|trade|goes|down

ADVERB very|I|quickly|tentatively|down

PREPOSITION to|from|behind|below|between|at|of|in

CONJUCTION if|then|and|but|or|until

ADJECTIVE his|young|long|two

NOUN boy|girl|John|letter|teacher|house|street|end|center|ring|boxers|gloves|punches|champ

PRONOUN me|she|who|his

DETERMINER a|the

INTERJECTION Oh!|Wow!|Oops!

%%

[ \n\t]+ ;

{VERB} { printf("%s\t==> VERB\n",yytext); }

{ADVERB} { printf("%s\t==> ADVERB\n",yytext); }

{PREPOSITION} { printf("%s\t==> PREPOSITION\n",yytext); }

{CONJUCTION} { printf("%s\t==> CONJUCTION\n",yytext); }

{ADJECTIVE} { printf("%s\t==> ADJECTIVE\n",yytext); }

{NOUN} { printf("%s\t==> NOUN\n",yytext); }

{PRONOUN} { printf("%s\t==> PRONOUN\n",yytext); }

{DETERMINER} { printf("%s\t==> DETERMINER\n",yytext); }

{INTERJECTION} { printf("%s\t==> INTERJECTION\n",yytext); }

. ;

%%

int yywrap(void){ }

int main()

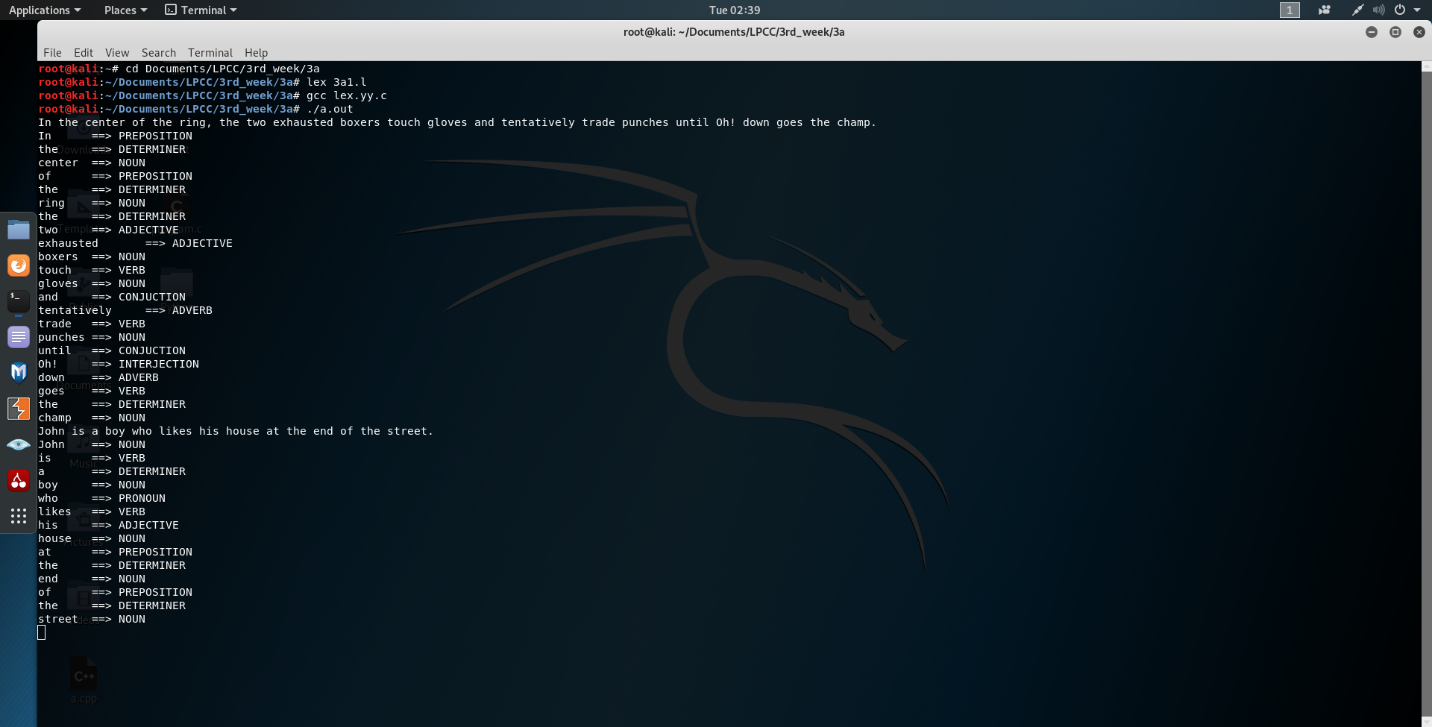
{

yylex();

return 0;

}

**OUTPUT :-**



**3C:-**

%{

%}

DIGIT [0-9]

NUMBER [-+]?[0-9]\*|[-+]?[0-9]\*[.][0-9]+

TEXT [A-Za-z]

PREPROCESSOR #include|#define|#undef

HEADERFILE [a-z]+\.h

KEYWORDS auto|break|case|const|continue|default|enum|extern|goto|register|return|signedsizeof|static|struct|switch|typedef|union|unsigned|volatile

INBUILT printf|scanf|clrscr\(\)|getch\(\)|main\(\)

DATATYPE boolean|protected|double|byte|int|short|void|char|long|float

CONDITIONAL if|switch|else|case

ITERATIVE while|for|do

STRINGLITERAL \"[^\n]\*\"

IDENTIFIER [A-Za-z$\_]({DIGIT}|{TEXT}|\_|$)\*

ARITH\_OP "+"|"-"|"/"|"%"|"\*"

LOGICAL\_OP "&&"|"||"|"!"|"!="

REL\_OP "<"|">"|"<="|">="|"=="

UNARY "++"|"--"

SC "."|"{"|"}"|","|":"|"'"|";"|"("|")"

FORMATSPECIFIER %d|%c|%s|%f

%%

[ \n\t]+ ;

{NUMBER} { printf("%s\t==> NUMBER\n",yytext); }

{PREPROCESSOR} { printf("%s\t==> PREPROCESSOR\n",yytext); }

{HEADERFILE} { printf("%s\t==> HEADERFILE\n",yytext); }

{KEYWORDS} { printf("%s\t==> KEYWORDS\n",yytext); }

{INBUILT} { printf("%s\t==> INBUILTFUNC\n",yytext); }

{DATATYPE} { printf("%s\t==> DATATYPE\n",yytext); }

{CONDITIONAL} { printf("%s\t==> CONDITIONAL\n",yytext); }

{ITERATIVE} { printf("%s\t==> ITERATIVE\n",yytext); }

{IDENTIFIER} { printf("%s\t==> IDENTIFIER\n",yytext); }

{ARITH\_OP} { printf("%s\t==> ARITHMETIC OPERATOR\n",yytext); }

{LOGICAL\_OP} { printf("%s\t==> LOGICAL OPERATOR\n",yytext); }

{REL\_OP} { printf("%s\t==> RELATIONAL OPERATOR\n",yytext); }

{UNARY} { printf("%s\t==> UNARY OPERATOR \n",yytext); }

{SC} { printf("%s\t==> PUNCTUATION\n",yytext); }

{FORMATSPECIFIER} { printf("%s\t==> FORMAT SPECIFIER\n",yytext); }

{STRINGLITERAL} { printf("%s\t==> STRINGLITERAL\n",yytext); }

"=" { printf("%s\t==> ASSIGNMENT OP\n",yytext); }

. ;

%%

int yywrap(void){ }

int main()

{

yylex();

return 0;

}

**OUTPUT:-**

