**LPCC ASSIGNMENT-3(WEEK-1)**

****

**SUBMITTED TO:**

**PROF. PALLAVI REGE**

**VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE**

**COMPUTER ENGINEERING DEPARTMENT**

**BY:**

**NAME: ABHISHEK MORE**

**G.R No.: 21810033**

**ROLL NO.: 323036**

**CLASS: T.Y COMP**

**BATCH: COMP C2**

**ASSIGNMENT-3(WEEK-1)**

**AIM:**

1. Write a program to implement a lexical analyzer for parts of speech, Using LEX.
   * 3A: For parts of speech for subset of ENGLISH language without using SYMBOL TABLE
   * 3C: Write Lexical analyser without using SYMBOL TABLE for subset of ‘C’ programming language

**SOURCE CODE:**

**3A:**

%{  
%}  
%%  
is|am|are|were|was|be|being|been|do|does|did|will|would|should|can|could|has|have|had|go       { printf("%s: is a verb\n", yytext); }  
  
very|simply|gently|quietly|calmly|angrily   { printf("%s: is an adverb\n", yytext); }  
  
to|from|behind|above|below|between    { printf("%s: is a preposition\n", yytext); }  
  
if|then|and|but|or       { printf("%s: is a conjunction\n", yytext); }  
  
their|my|your|his|her|its      { printf("%s: is a adjective\n", yytext); }  
  
I|you|he|she|we|they       { printf("%s: is a pronoun\n", yytext); }  
  
Oh!|Wow!|Oops!   { printf("%s: is a interjection\n", yytext); }  
%%  
int yywrap()  
{  
return 1;  
}  
  
void main()  
{  
yylex();  
yywrap();  
}

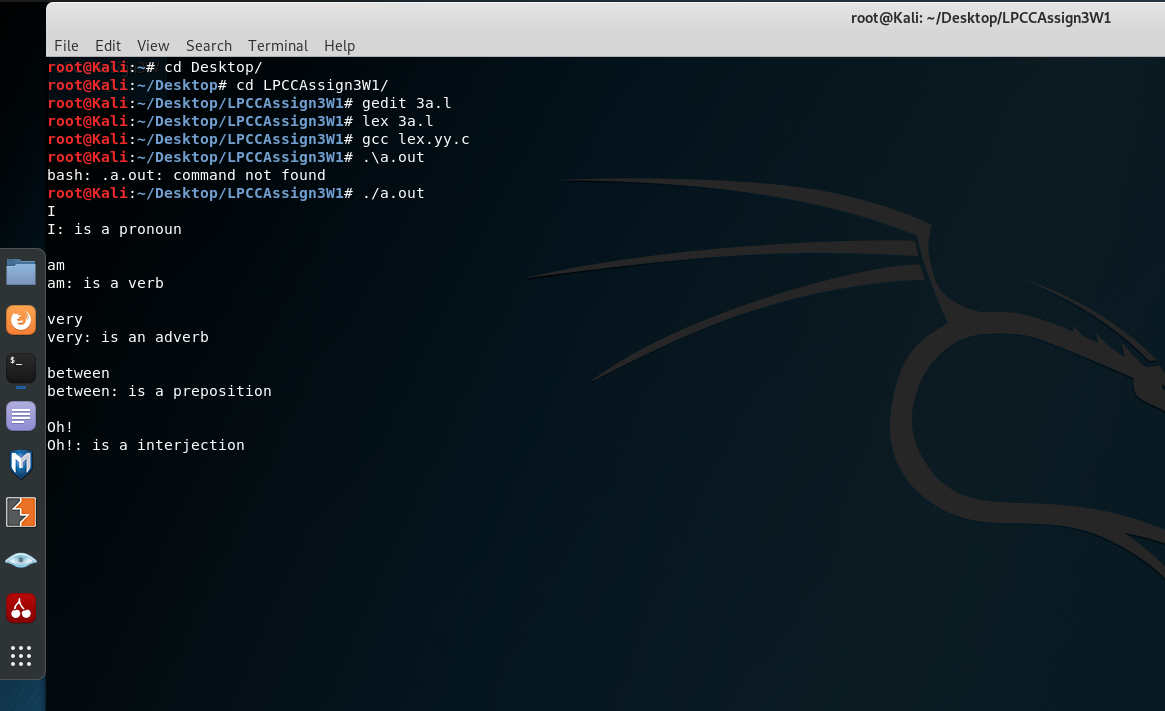
**3C:**

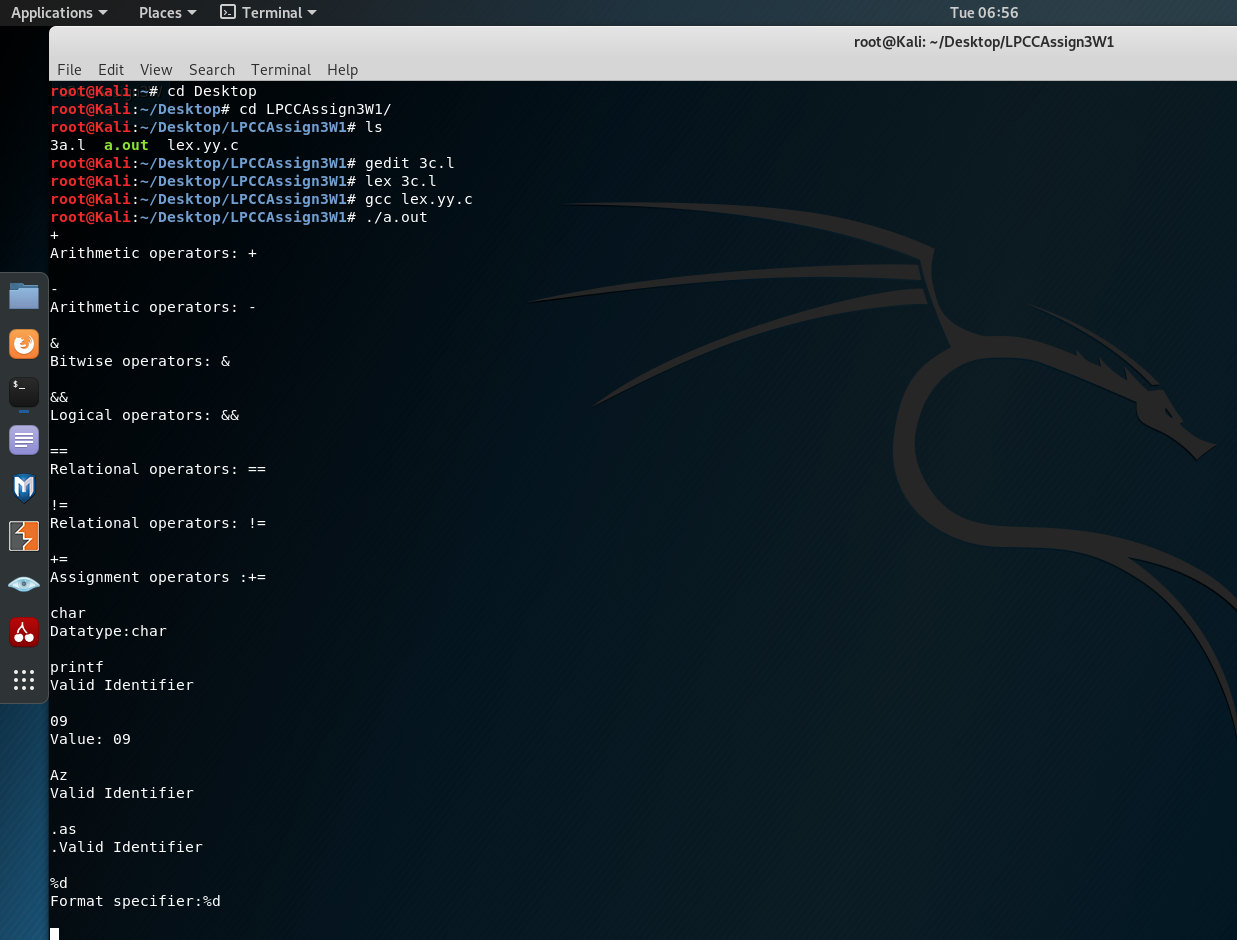
%{  
%}  
%%  
"+"|"-"|"\*"|"/"|"%" {printf("Arithmetic operators: %s\n",yytext);}  
"=="|"!="|">"|"<"|">="|"<=" {printf("Relational operators: %s\n",yytext);}  
"&"|"|"|"^"|"~"|">>"|"<<" {printf("Bitwise operators: %s\n",yytext);}  
"&&"|"||"|"!" {printf("Logical operators: %s\n",yytext);}  
"="|"+="|"-="|"\*=" {printf("Assignment operators :%s\n",yytext);}  
"char"|"Char"|"int"|"Int"|"string"|"String"|"Float"|"float" {printf("Datatype:%s\n",yytext);}  
[-+]?[0-9]\* {printf("Value: %s\n",yytext);}  
"%d"|"%c"|"%s"|"%f" {printf("Format specifier:%s\n",yytext);}  
[a-zA-Z\_][a-zA-Z0-9\_]\* { printf("Valid Identifier\n");}  
  
%%  
int yywrap()  
{  
return 1;  
}  
  
void main()  
{  
yylex();  
yywrap();

}

**OUTPUT:**

**3A:**

****

**3C:**

**CONCLUSION:**

Successfully implemented a lexical analyser using lex for part of speech without using Symbol table and also implemented lexical analyser for identifying the special characters and operators.