

# ASSIGNMENT 3 WEEK-1

Name:Pranav Sarda

Gr No: 21810773

Roll No: 321047

Div: A-2

## 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||quickly|tentatively|down
PREPOSITION to|from|behind|below|between|at|of|in
CONJUNCTION 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); }
{CONJUNCTION} { printf("%s\t==> CONJUNCTION\n",yytext); }
{ADJECTIVE} { printf("%s\t==> ADJECTIVE\n",yytext); }
{NOUN}      { printf("%s\t==> NOUN\n",yytext); }
{PRONOUN}   { printf("%s\t==> PRONOUN\n",yytext); }
```

# ASSIGNMENT 3 WEEK-1

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

```
%%
```

```
int yywrap(void){ }
```

```
int main()
```

```
{
```

```
    yylex();
```

```
    return 0;
```

```
}
```

# ASSIGNMENT 3 WEEK-1

## OUTPUT :-

```
root@kali:~# cd Documents/LPCC/3rd_week/3a
root@kali:~/Documents/LPCC/3rd_week/3a# lex 3a1.l
root@kali:~/Documents/LPCC/3rd_week/3a# gcc lex.yy.c
root@kali:~/Documents/LPCC/3rd_week/3a# ./a.out
In the center of the ring, the two exhausted boxers touch gloves and tentatively trade punches until Oh! down goes the champ.
In      ==> PREPOSITION
the     ==> DETERMINER
center ==> NOUN
of      ==> PREPOSITION
the     ==> DETERMINER
ring    ==> NOUN
the     ==> DETERMINER
two     ==> ADJECTIVE
exhausted ==> ADJECTIVE
boxers  ==> NOUN
touch   ==> VERB
gloves  ==> NOUN
and     ==> CONJUNCTION
tentatively ==> ADVERB
trade   ==> VERB
punches ==> NOUN
until   ==> CONJUNCTION
Oh!     ==> INTERJECTION
down    ==> ADVERB
goes    ==> VERB
the     ==> DETERMINER
champ   ==> NOUN
John is a boy who likes his house at the end of the street.
John    ==> NOUN
is      ==> VERB
a       ==> DETERMINER
boy     ==> NOUN
who     ==> PRONOUN
likes   ==> VERB
his     ==> ADJECTIVE
house   ==> NOUN
at      ==> PREPOSITION
the     ==> DETERMINER
end     ==> NOUN
of      ==> PREPOSITION
the     ==> DETERMINER
street  ==> NOUN
[]
C++
2022
```

# ASSIGNMENT 3 WEEK-1

### 3C:-

```
%{  
DIGIT          [0-9]  
NUMBER         [-+]?[0-9]*[-+]?[0-9]*.[0-9]+  
TEXT           [A-Za-z]  
PREPROCESSOR   #include|#define|#undef  
HEADERFILE     [a-z]+\.  
KEYWORDS       auto|break|case|const|continue|default|enum|extern|goto|register|return|signed|sizeof|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  
%%  
[ \\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;  
}
```

# ASSIGNMENT 3 WEEK-1

}

## OUTPUT:-

```
File Edit View Search Terminal Help
root@kali:~/Documents/LPCC/3rd_week/3a# cd ..
root@kali:~/Documents/LPCC/3rd_week# cd 3c
root@kali:~/Documents/LPCC/3rd_week/3c# lex 3c1.l
root@kali:~/Documents/LPCC/3rd_week/3c# gcc lex.yy.c
root@kali:~/Documents/LPCC/3rd_week/3c# ./a.out
#include <stdio.h>
Downloads a.out
#include ==> PREPROCESSOR
<stdio.h> ==> HEADERFILE
void main()
void ==> DATATYPE
main() ==> INBUILTFUNC
{ Templates program.c
{ ==> PUNCTUATION
int num;
int ==> DATATYPE
num ==> IDENTIFIER
; ==> PUNCTUATION
printf("Enter a number: \n");
printf ==> INBUILTFUNC
( ==> PUNCTUATION
"Enter a number: \n" ==> STRINGLITERAL
) ==> PUNCTUATION
; ==> PUNCTUATION
scanf
scanf ==> INBUILTFUNC
%d
%d ==> FORMAT SPECIFIER
if (num > 0)
if ==> CONDITIONAL
( ==> PUNCTUATION
num ==> IDENTIFIER
> ==> RELATIONAL OPERATOR
0 ==> NUMBER
) ==> PUNCTUATION
printf("%d is a positive number \n", num);
printf ==> INBUILTFUNC
( ==> PUNCTUATION
"%d is a positive number \n" ==> STRINGLITERAL
, ==> PUNCTUATION
num ==> IDENTIFIER
) ==> PUNCTUATION
; ==> PUNCTUATION
}
} ==> PUNCTUATION
C++
3.cpp
```