COMPILER DESIGN LAB

WEEK 4 (8.1.19) – EXERCISE

SET – A

1. Write a Lex program for the following

a. All strings of a’s and b’s that contain at least two b's.
b. All strings of lowercase consonants.
c. All strings of digits with no repeated digits.
d. All strings of lowercase letters in which the letters in are in ascending lexicographic order.
e. The characters that can appear at the end of a legitimate English sentence (e.g. , exclamation point) .
f. Find the token and its count from the given

```
c = a++    +    ++b
```

g. Identify the tokens in the given input statement

```
def f(x):
if x >= 1:
 return x * x
else:
 return x
print 3
```

SET – B

1. Write a Lex program for the following

a. All strings of a’s and b’s that contain at most two b's.
b. All strings of a’s and b’s that contain just two or three b`s
c. All strings of lowercase letters that contain the five vowels in order.
d. Comments, consisting of a string surrounded by /* and */, without an intervening */, unless it is inside double-qoutes(“)

Input:

```
/*** Definition section ***/

%{      /* C code to be copied verbatim */
%}

/* This tells flex to read only one input file */

%%         /*** Rules section ***/
         /* [0-9]+ matches a string of one or more digits */
[0-9]+  { /* yytext is a string containing the matched text. */
             }
.|\n    {   /* Ignore all other characters. */   }
%%

/*** C Code section ***/
```

e. Find the tokens and its count from the given

```
c = ++a   +   ++b
```

f. Identify the tokens in the given input statement

```
print  ( 3 + x *2)
def f(x): return 3
```

g. Find or search a pattern matching using Grep syntax to recognize the given input string is verb.