1) Christian McGovern, Lab1, 1/19/2018

2) I added new rule below the [a-zA-Z]+ lgths[yyleng]++; to count numbers. The new line was [0-9]+ countnums++. I also created int countnums and printed below Dr. Coopers for loop.

3)

```
/*  This lex routine uses a counting array to match alphabeticstrings
    and make a frequency count.

    The real item to notice is that yywrap() is called at EOF and then is run
    to do what we need to do.  yywrap() returns true when we have a successful
    end to the program.  We may want to return false (0) if we want to lexing process
    to fail

   Shaun Cooper
   January 2015
 */
 /*
            Christian McGovern
            Lab1, 1/19/2018
            Changes to lines: 20, 23, and 37
  */
    int lgths[100];
    int countnums = 0;//int added to count numbers
%%
[a-zA-Z]+   lgths[yyleng]++;
```

```
[0-9]+          countnums++;// rule/line added to count numbers
.       |
\n      ;

%%
yywrap()
{
        int i;
        printf("Length  No. words  \n");
        for (i=1; i<100; i++) {
                if (lgths[i] > 0) {
                        printf("%5d%10d\n",i,lgths[i]);
                }
        }
        printf("No. Numbers: %15d\n", countnums);//line added to print No. Numbers
        return(1);
}
main()
{  yylex();
}
```

4)

```
#Christian McGovern
#Lab1
```

#1/19/2018

lex.yy.c: wordlengthlab1.l

lex wordlengthlab1.l

gcc -o wordlength lex.yy.c

5)

```
mcgovern@Christian:/mnt/c/Users/Christian/Desktop/Google_Drive/Course-Work/NMSU-Compilers/lab1$ ./wordlength < /etc/passwd
Length  No. words
     1        25
     2         2
     3        64
     4        47
     5        18
     6         9
     7        22
     8         2
     9         5
    10         1
    11         1
No. Numbers:             50
```