Long Tran
Lab 1
January 24, 2020
Changes made to the code:
+ Add a new int variable to count the number of numbers
+ Add new lex directive that finds numbers and increase the count when it has found one.
+ Add a print statement in yywrap() to print out the number of numbers

wordlengthlab1.l

```
%{
        /*  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
  */

  /* modified wordlengthlab1.l

          This lex routine now uses a counting array to match alphabeticstrings and
          make a frequency count. It also matches numbers and use a counting
          variale to count the number of numbers when it found one.

          Long Tran
          January 24, 2020

  */

        // a counting variable to count the number of lengths of alphabetic strings
        int lgths[100];

        // a counting varibale to count the number of numbers
        int count = 0;
%}
%%
[a-zA-Z]+       {lgths[yyleng]++;/*lex directive to match alphabeticstrings and
increase the count of the length of the found string*/}
[0-9]+          {count++;/* lex directive to find numbers and when it has found one,
it will increase the count of numbers */}
.       |
\n      ;
%%
```

```
// called at the end of file
int yywrap()
{
	// print out the number of different lengths of the strings found in the
	// input
	printf("Length  No. words\n");

	// traverse throught the array and print out the value in it
	int i;
	for (i=1; i<100; i++) {
		if (lgths[i] > 0) {
			printf("%5d%10d\n",i,lgths[i]);
		} // end if
	} // end for i

	// print out the number of numbers
	printf("number of numbers is %d\n", count);

	// return 1 because I don't want to scan more input
	return(1);
}

// main function
int main()
{
	// call yylex() to start the scanning of the input
	yylex();
}
```

Makefile

```
#
# Makefile for lab 1
# Long Tran
# January 24, 2020
#
all:	lab1
lab1:	lex.yy.c
	gcc -o lab1 lex.yy.c
lex.yy.c:	wordlengthlab1.l
	lex wordlengthlab1.l
```

*** screenshot of output ***

```
ltran@lappy12:~/CS370/Lab1> make
lex wordlengthlab1.l
gcc -o lab1 lex.yy.c
ltran@lappy12:~/CS370/Lab1> ./lab1
a abc ab123cd 983 10923 1233950
Length  No. words
    1         1
    2         2
    3         1
number of numbers is 4
ltran@lappy12:~/CS370/Lab1>
```