Compiler Design Report

Shuvayan Ghosh Dastidar
001810501044
BCSE-III
Third Year Second Semester
2021

Assignment 1

Lex is a tool that generates Lexical Analyzer from the regular expressions for a given language.

Problem Statement

Lex is a tool that generates Lexical Analyzer from the regular expressions for a given language. There are several tutorials available on the Internet for learning lex.

Go through the tutorials and implement the followings.

Write a lex file to count the number of lines, words, and characters in the input.

Write a lex file to count the number of numbers appearing in the input. Count the number of integers (without a decimal) separately from the number of floating point numbers (with a decimal, and at least one digit on either side of the decimal).

Write a lex file to count the number of words in an input text that start with a vowel.

You may find solutions to the above problems on the Internet. Just copying from those sources without understanding their working will not be helpful for your learning.

Question 1

Write a lex file to count the number of lines, words, and characters in the input.

count.lex

```
%{
int num_lines=0, num_chars =0, num_words=0;
%}
%%

\n      num_lines++;num_chars++;
([a-zA-Z0-9])+ {num_words++; num_chars +=strlen(yytext);}
.       num_chars++;

%%

int yywrap(void)
{
 return 1;
}

int main(){
  yylex();
  printf("Number of lines : %d\n", num_lines);
  printf("Number of characters : %d\n", num_chars);
  printf("Number of words : %d\n", num_words);
}
```

The program parses the given input and counts the number of lines, characters and words from the input.

Input : sample.txt

This is a sample txt file.
It contains two lines.
It contains some words and characters.
Bye Human.

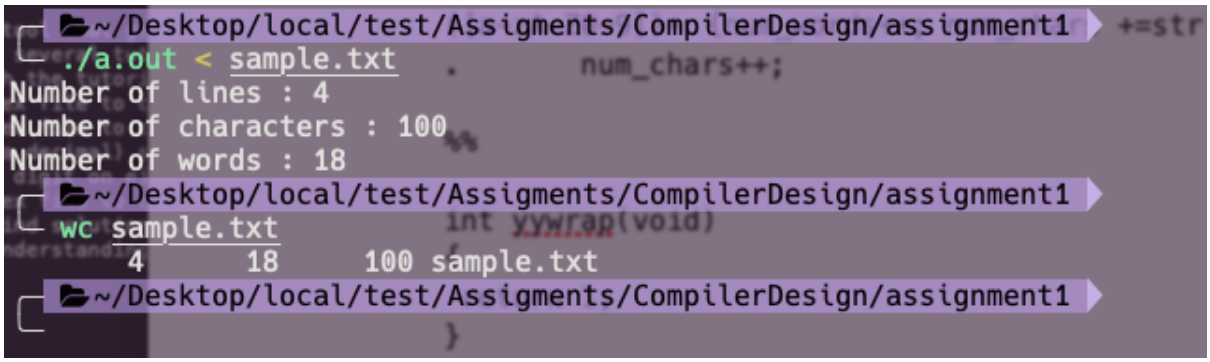
Output:

$ flex count.lex

$ gcc lex.yy.c

$ ./a.out < sample.txt

Output screenshot showing the validation with the unix wc command which performs the same operation.



Question 2

Write a lex file to count the number of numbers appearing in the input. Count the number of integers (without a decimal) separately from the number of floating point numbers (with a decimal, and at least one digit on either side of the decimal).

count_num.lex

```
%{
int num_dec=0, num_float=0;
%}

%%
\n   ;
[0-9]+\.[0-9]+ {printf("Floating point : %s\n", yytext);
  num_float++; }
[0-9]+ {printf("Decimal : %s\n", yytext); num_dec++;}
. ;



%%

int yywrap(void)
{
 return 1;
}

int main(){
  yylex();
  printf("Number of decimal numbers: %d\n", num_dec);
  printf("Number of floating numbers : %d\n", num_float);
}
```

The program parses the given input and counts the number of integers and floating point numbers from the given input based on the regular expressions inside %% .

Input : num.txt

5456 343.343 dkfk 343 43.34 43.

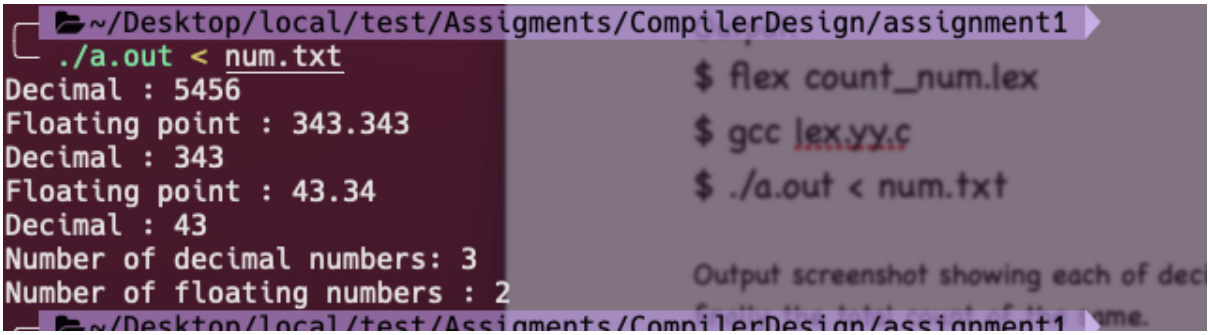
Output:

$ flex count_num.lex

$ gcc lex.yy.c

$ ./a.out < num.txt

Output screenshot showing each of decimal and floating point numbers as it parses and finally the total count of the same.



Question 3

Write a lex file to count the number of words in an input text that start with a vowel.

count_vowel.lex

```
%{
int num_words=0;
%}

%%
\n   ;
[aeiouAEIOU][a-zA-Z0-9.]*     {num_words++; printf("%s\n",
  yytext);}
[a-zA-Z0-9(^aeiouAEIOU)][a-zA-Z0-9]* ;
. ;



%%

int yywrap(void)
{
 return 1;
}

int main(){
  yylex();
  printf("Number of words that start with a vowel: %d\n",
  num_words);
}
```

The program parses the given input and counts the number of words starting with a vowel and skips any other character from the given input based on the regular expressions inside %% .

Input : sample.txt

This is a sample txt file.
It contains two lines.
It contains some words and characters.
Bye Human.

Output:

$ flex count_vowel.lex

$ gcc lex.yy.c

$ ./a.out < sample.txt

Output screenshot showing the words starting with a vowel and finally the total count of them.