

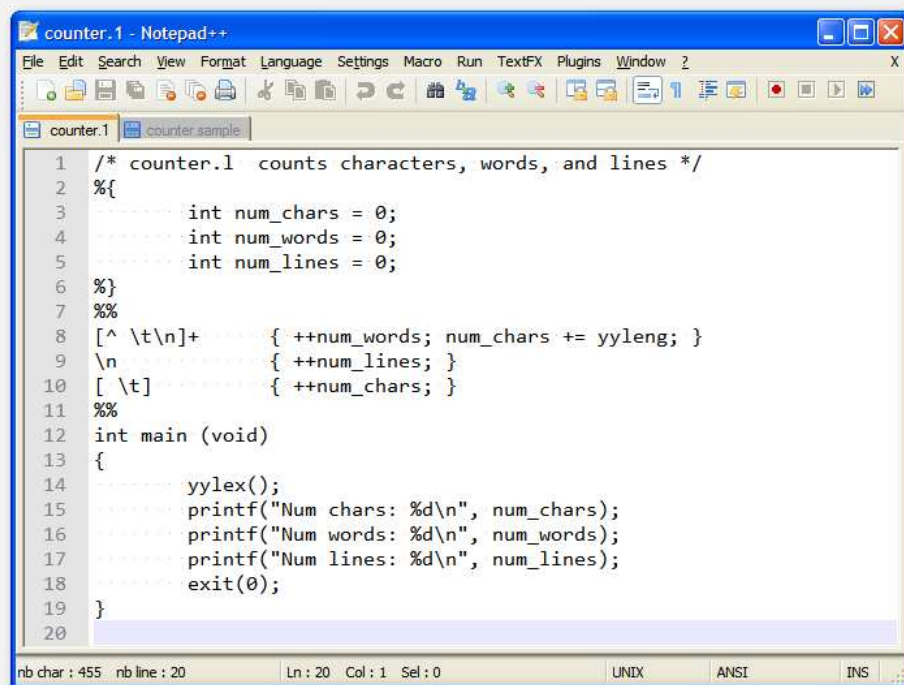
TA: Jade Cheng
ICS 312
Homework Solution #20
Due Date: November 14, 2009

Lex Homework: Counting Vowel-Consonant Pairs

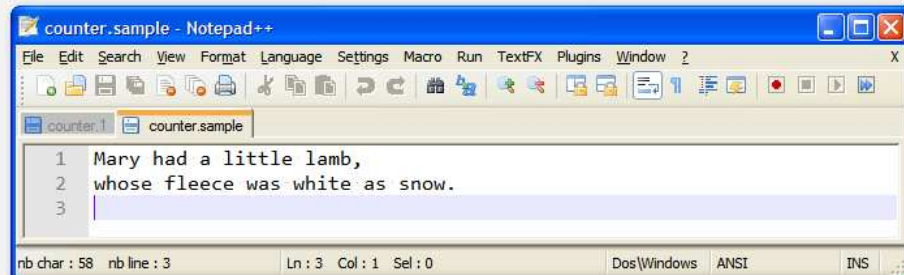
Part I: Repeat the instructed procedures to count numbers of chars, words, and lines.

Procedure: Note that the new uhunix server has lex and yacc installed, so you can do the homework on uhunix if you like. The following procedure was done on uhunix.

First, create two files, counter.1 and counter.sample. The contents of these files are shown below:

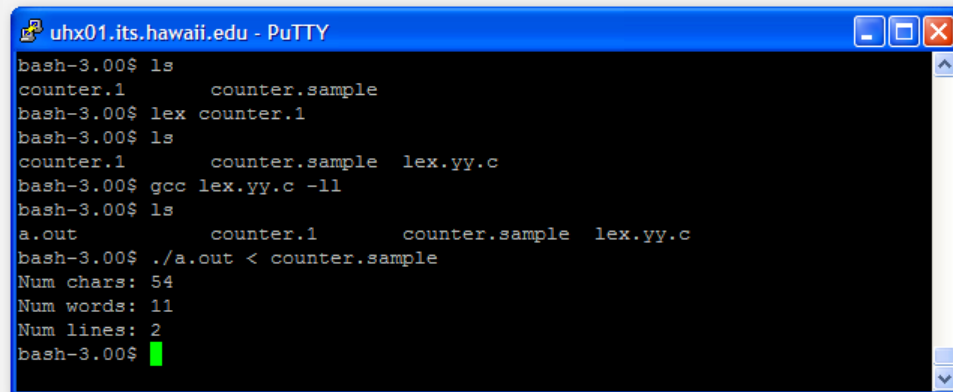


```
1 /* counter.1 counts characters, words, and lines */
2 %{
3     int num_chars = 0;
4     int num_words = 0;
5     int num_lines = 0;
6 }%
7 %%
8 [^ \t\n]+ { ++num_words; num_chars += yyleng; }
9 \n { ++num_lines; }
10 [ \t] { ++num_chars; }
11 %%
12 int main (void)
13 {
14     yylex();
15     printf("Num chars: %d\n", num_chars);
16     printf("Num words: %d\n", num_words);
17     printf("Num lines: %d\n", num_lines);
18     exit(0);
19 }
20
```



```
1 Mary had a little lamb,
2 whose fleece was white as snow.
3
```

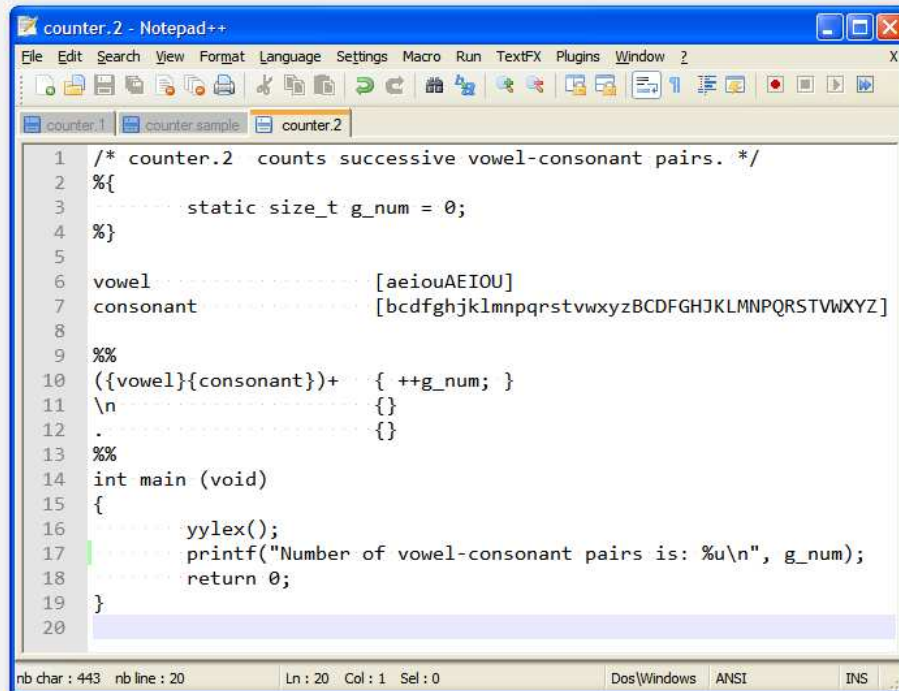
Transfer your files to uhunix, and then use `lex` to generate the C file, `lex.yy.c`. Compile `lex.yy.c` with `gcc` using the `-ll` flag to link the `lex` library, `libl.so`, with your program. This generates an executable file named `a.out`. Then execute `a.out`, redirecting `counter.sample` for standard input.



```
uhx01.its.hawaii.edu - PuTTY
bash-3.00$ ls
counter.1      counter.sample
bash-3.00$ lex counter.1
bash-3.00$ ls
counter.1      counter.sample  lex.yy.c
bash-3.00$ gcc lex.yy.c -ll
bash-3.00$ ls
a.out         counter.1      counter.sample  lex.yy.c
bash-3.00$ ./a.out < counter.sample
Num chars: 54
Num words: 11
Num lines: 2
bash-3.00$
```

Part II: Alter the program so that it instead counts sequences of successive vowel-consonant pairs.

Procedure: Create a `counter.2` file that defines a grammar that will count sequences of successive vowel-consonant pairs. The contents of the file are shown below:



```
counter.2 - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
counter.1 counter.sample counter.2
1 /* counter.2 counts successive vowel-consonant pairs. */
2 %{
3     static size_t g_num = 0;
4 }
5
6 vowel ..... [aeiouAEIOU]
7 consonant ..... [bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQrstvwXYZ]
8
9 %%
10 ({vowel}{consonant})+ { ++g_num; }
11 \n ..... {}
12 . ..... {}
13 %%
14 int main (void)
15 {
16     yylex();
17     printf("Number of vowel-consonant pairs is: %u\n", g_num);
18     return 0;
19 }
20
nb char : 443 nb line : 20 Ln : 20 Col : 1 Sel : 0 Dos/Windows ANSI INS
```

Transfer your `counter.2` file to uhunix, and then use `lex` to generate the C file, `lex.yy.c`. Compile `lex.yy.c` with `gcc` using the `-ll` flag to link the `lex` library, `libl.so`, with your program. This generates an executable file named `a.out`. Then execute `a.out`, redirecting `counter.sample` for standard input.

```
uhx01.its.hawaii.edu - PuTTY
bash-3.00$ ls
counter.1      counter.2      counter.sample
bash-3.00$ lex counter.2
bash-3.00$ ls
counter.1      counter.2      counter.sample  lex.yy.c
bash-3.00$ gcc lex.yy.c -ll
bash-3.00$ ls
a.out          counter.2      lex.yy.c
counter.1      counter.sample
bash-3.00$ ./a.out < counter.sample
Number of vowel-consonant pairs is: 10
bash-3.00$ █
```