Assignment 1(date: 2/1/2018)

Submitted By: LIPSA CHHOTRAY, Roll no: 115cs0203

A1.1 Design a scanner using flex to count the number of characters words, lines and punctuations in a given string.

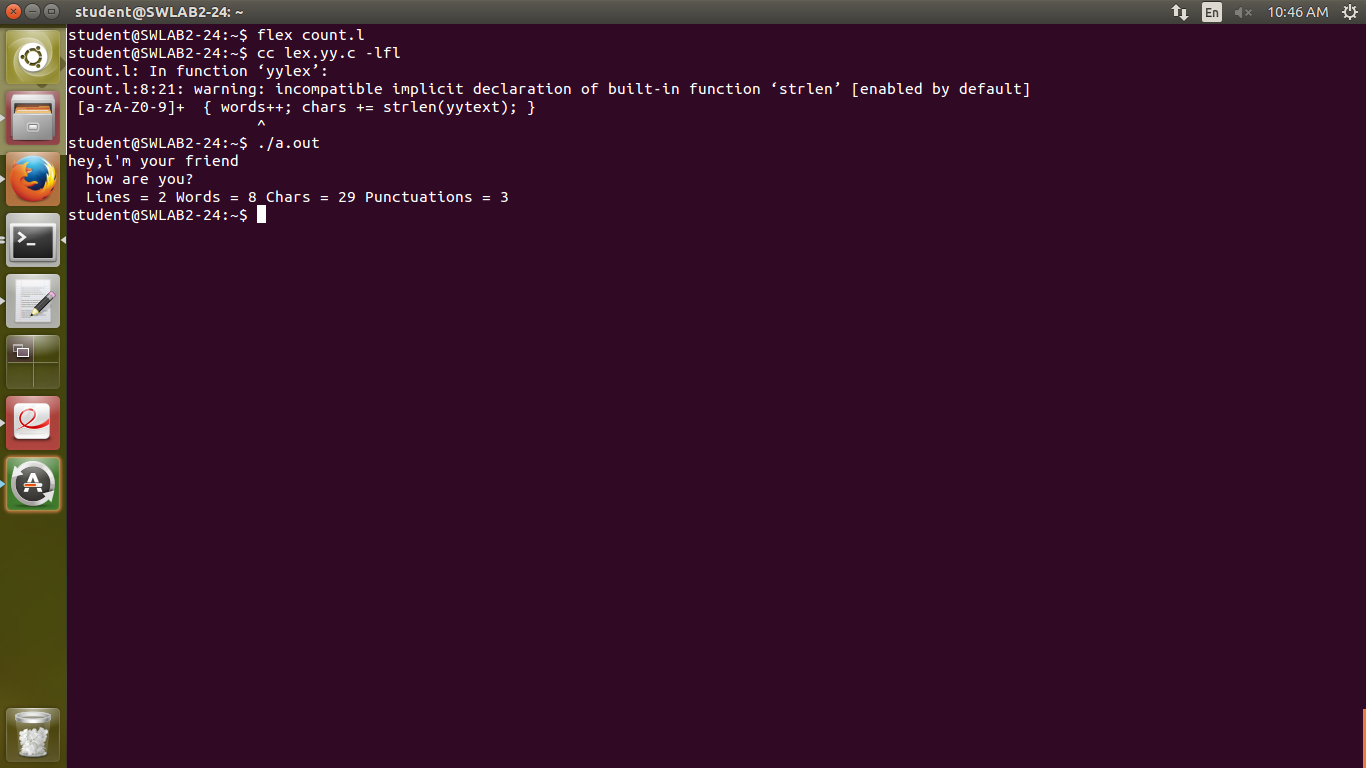
Symbol set: [a-Za-z0-9]+

Punctuation symbol: [. , : ; ‘ ” ? ]

Program:

%{  
int chars = 0;  
int words = 0;  
int lines = 0;  
int puncs = 0;  
%}  
%%  
[a-zA-Z0-9]+  { words++; chars += strlen(yytext); }  
\n         { chars++; lines++; }  
[.,;:?"\"""\'"]  { chars++; puncs++;}  
  
%%  
main(int argc, char \*\*argv)  
{  
  yylex();  
  printf("Lines = %d Words = %d Chars = %d Punctuations = %d\n", lines, words, chars, puncs);  
}

**Output:**

****

A1.2 To convert British words to English words.

Program:

%%  
"colour" { printf(" -> color"); }  
"flavour" { printf(" -> flavor"); }  
"clever" { printf(" -> smart"); }  
"smart" { printf(" -> elegant"); }  
"metre" { printf(" -> meter"); }  
"litre" { printf(" -> litter"); }  
"programme" { printf(" -> program"); }  
"lift" { printf(" -> elevator"); }  
"football" { printf(" -> soccer"); }  
"curd" { printf(" -> yogurt"); }  
"truct" { printf(" -> lorry"); }  
"post" { printf(" -> mail"); }  
. { printf("%s", yytext); }  
%%  
main(int argc, char \*\*argv)  
{  
  yylex();  
}

Output:



Assignment 2(date: 9/1/2018)

Submitted By: LIPSA CHHOTRAY Roll no: 115cs0203

A 2.1 Write a flex application that will recognize valid identifier having symbols a-z, A-Z, 0-9 and underscore (\_) .Each identifier should not start with a symbol from (0-9).Any other symbol like decimal .Point/full stop (.) (,) semicolon (;) colon (:) + - \* / @ $ are considered as invalid symbols.

Program:

%{  
  
          
%}  
  
%%  
[A-Za-z\_][A-Za-z0-9\_]+ {printf("valid");}  
.\*    {printf("invalid");}   
%%  
  
int main(int argc, char \*\* argv)  
{  
        yylex();  
}

Output:



A2.2 Write a flex application that will recognize string of odd numbers of 0’s and even number of 1’s.

Program:

%{  
%}  
  
%%  
(0|1(00|11)\*(01|10))(00|11|(01|10)(00|11)\*(01|10))\* {printf("valid");}  
.\* {printf("invalid");}  
%%  
  
main(int argc,char \*\* argv)  
{  
        yylex();  
}

Output:

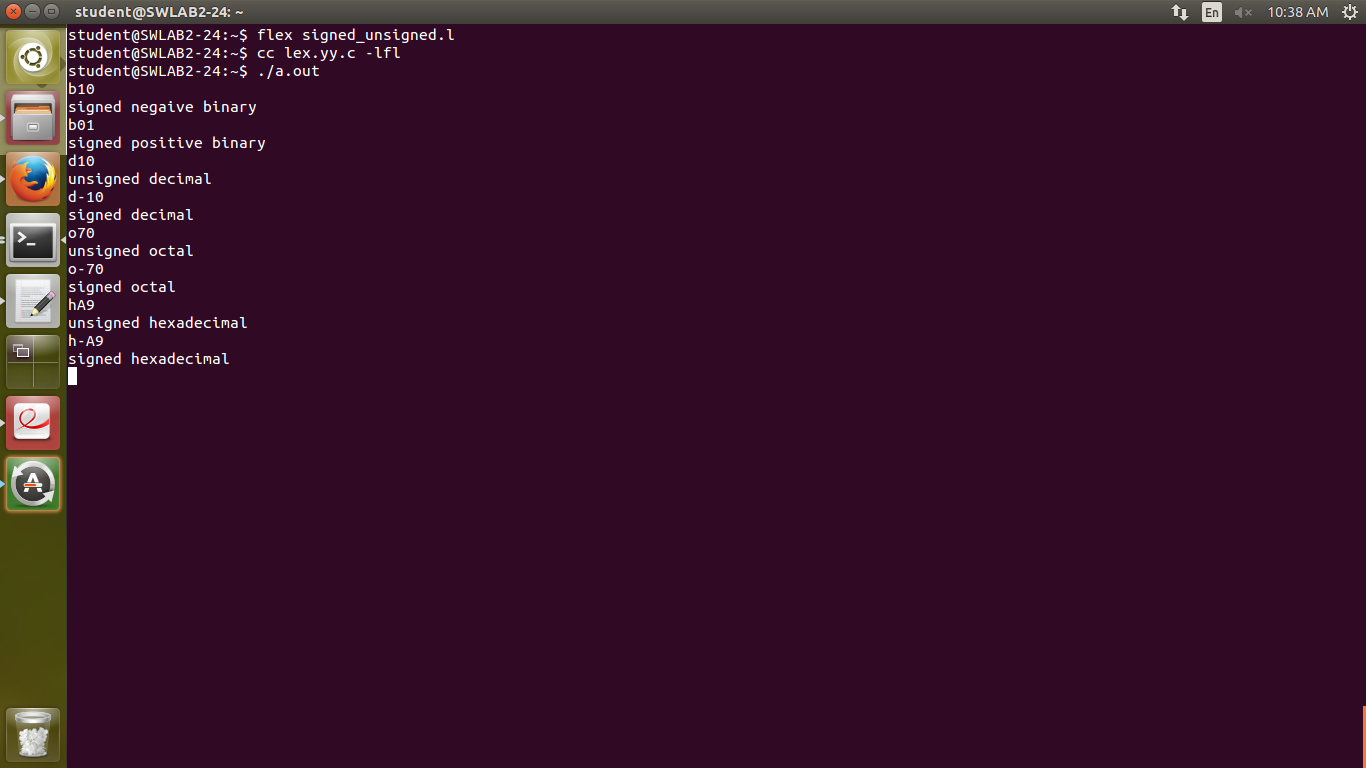


A2.3 Write flex application that will identify signed/unsigned integer and long integer constants in decimal, hexadecimal, binary and octal representations used in C language.

Program:

%{  
%}  
  
%%  
[b][0][0-1]\* {printf("signed positive binary");}  
[b][1][0-1]\* {printf("signed negaive binary");}  
[d][+-][0-9]\* {printf("signed decimal");}  
[d][0-9]\* {printf("unsigned decimal");}  
[o][+-][0-7]\* {printf("signed octal");}  
[o][0-7]\* {printf("unsigned octal");}  
[h][+-][0-9A-F]\* {printf("signed hexadecimal");}  
[h][0-9A-F]\* {printf("unsigned hexadecimal");}  
  
  
%%  
  
main(int argc ,char \*\* argv)  
{  
        yylex();  
}

Output:



A2.4 Write a flex program that will identify character and string constants as defined in C language.

Program:

%{  
%}  
  
%%  
['].['] {printf("char constant");}  
["].\*["]             {printf("string constant");}  
.\*         {printf("invalid constant");}  
%%  
  
main(int argc,char \*\* argv)  
{  
        yylex();  
}

Output:

