Lexical Analyzer

LEX TOOL

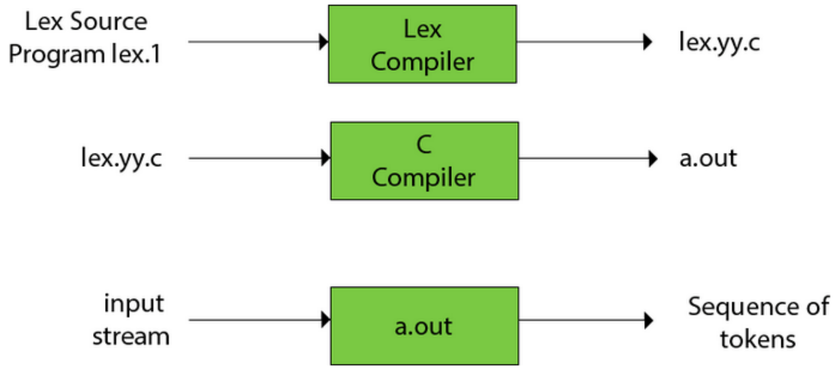
001810501069

Nikhil Badyal

BCSE III | JU

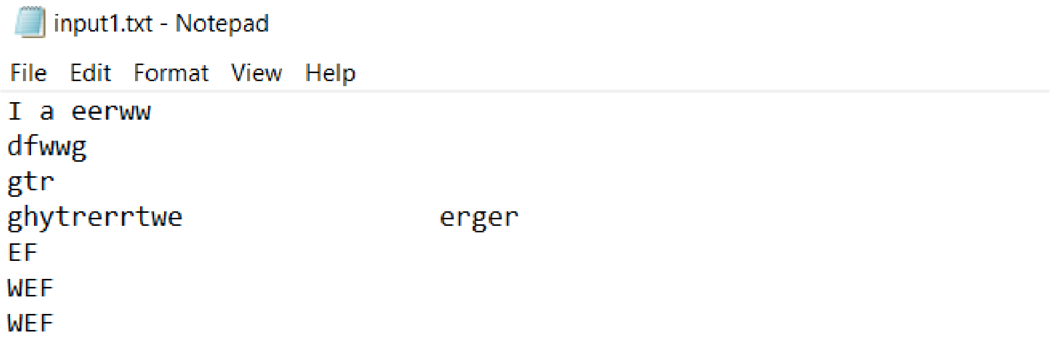
Lex is a computer program that generates lexical analyzers ("scanners" or "lexers").

- The lexical analyzer is a program that transforms an input stream into a sequence of tokens.
- It reads the input stream and produces the source code as output through implementing the lexical analyzer in the C program.



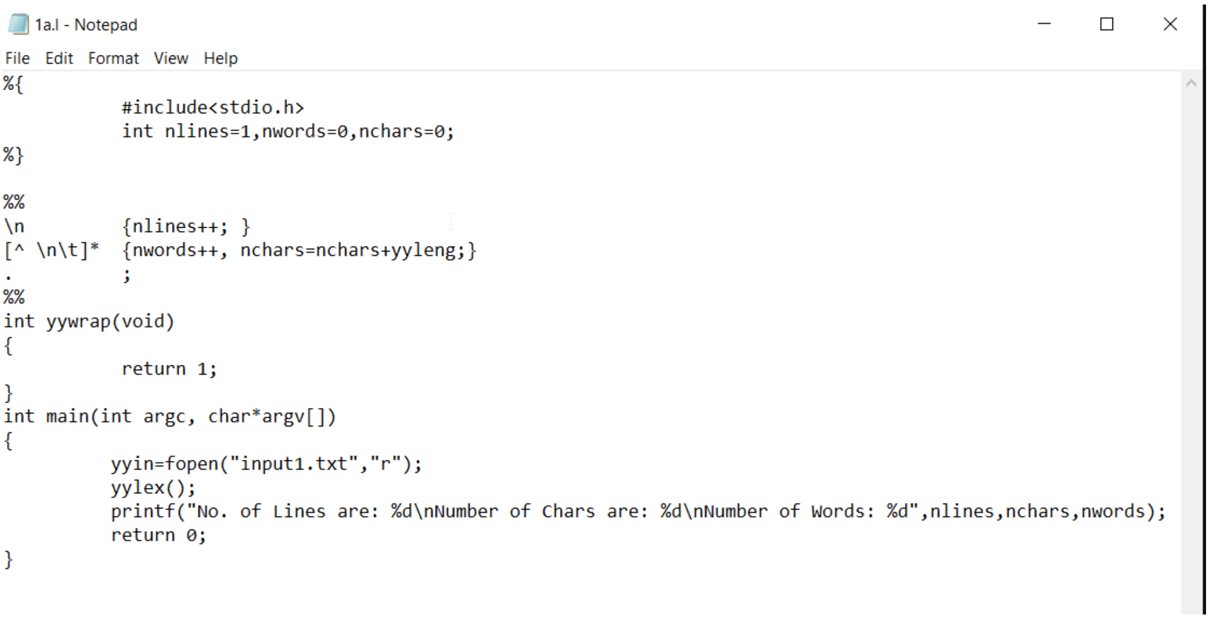
Program 1

Input Data



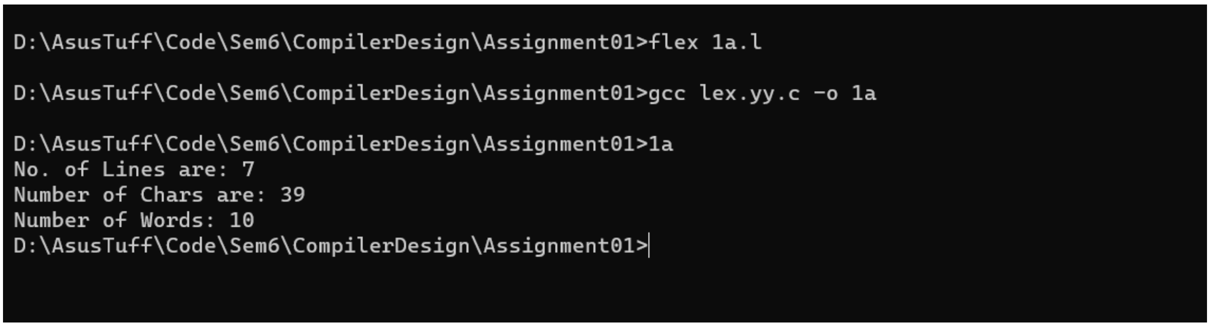
```
I a eerww
dfwwg
gtr
ghytrerrtwe                    erger
EF
WEF
WEF
```

Lex File



```
%{
           #include<stdio.h>
           int nlines=1,nwords=0,nchars=0;
%}

%%
\n         {nlines++; }
[^ \n\t]*  {nwords++, nchars=nchars+yyleng;}
.          ;
%%
int yywrap(void)
{
           return 1;
}
int main(int argc, char*argv[])
{
          yyin=fopen("input1.txt","r");
          yylex();
          printf("No. of Lines are: %d\nNumber of Chars are: %d\nNumber of Words: %d",nlines,nchars,nwords);
          return 0;
}
```

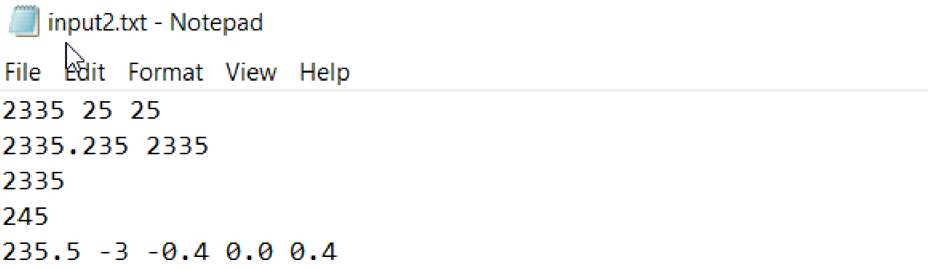
Output



```
D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>flex 1a.l

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>gcc lex.yy.c -o 1a

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>1a
No. of Lines are: 7
Number of Chars are: 39
Number of Words: 10
D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>
```

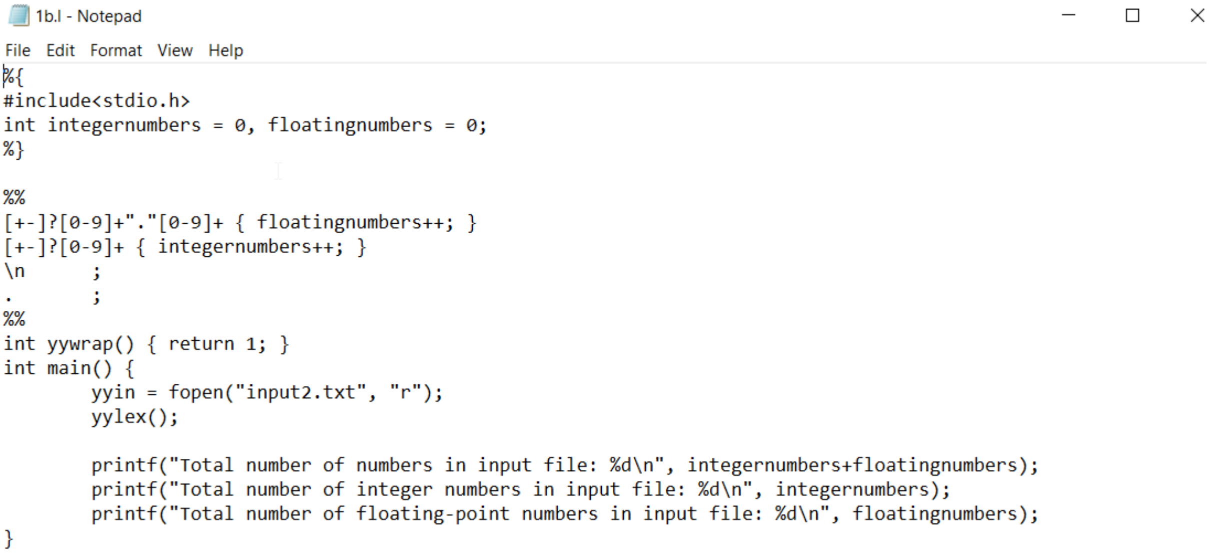
PROGRAM 2

Input Data



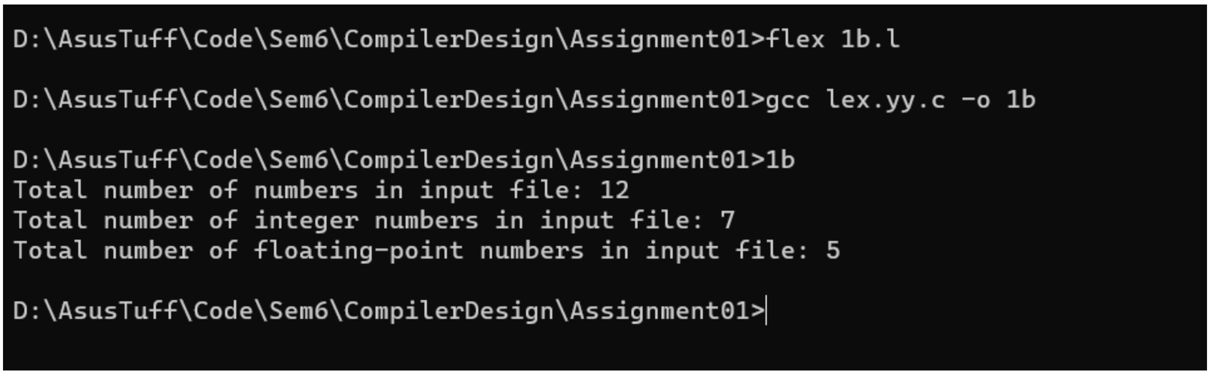
```
2335 25 25
2335.235 2335
2335
245
235.5 -3 -0.4 0.0 0.4
```

Lex File



```
%{
#include<stdio.h>
int integernumbers = 0, floatingnumbers = 0;
%}

%%
[+-]?[0-9]+"."[0-9]+ { floatingnumbers++; }
[+-]?[0-9]+ { integernumbers++; }
\n      ;
.       ;
%%
int yywrap() { return 1; }
int main() {
        yyin = fopen("input2.txt", "r");
        yylex();

        printf("Total number of numbers in input file: %d\n", integernumbers+floatingnumbers);
        printf("Total number of integer numbers in input file: %d\n", integernumbers);
        printf("Total number of floating-point numbers in input file: %d\n", floatingnumbers);
}
```

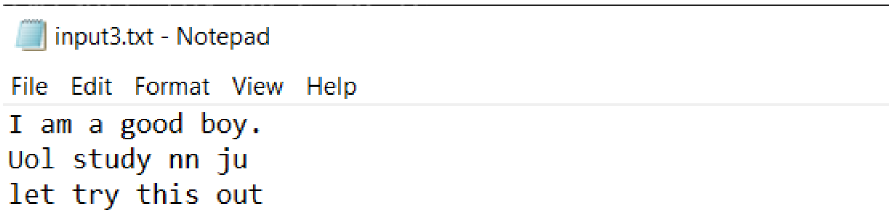
Output



```
D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>flex 1b.l

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>gcc lex.yy.c -o 1b

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>1b
Total number of numbers in input file: 12
Total number of integer numbers in input file: 7
Total number of floating-point numbers in input file: 5

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>
```

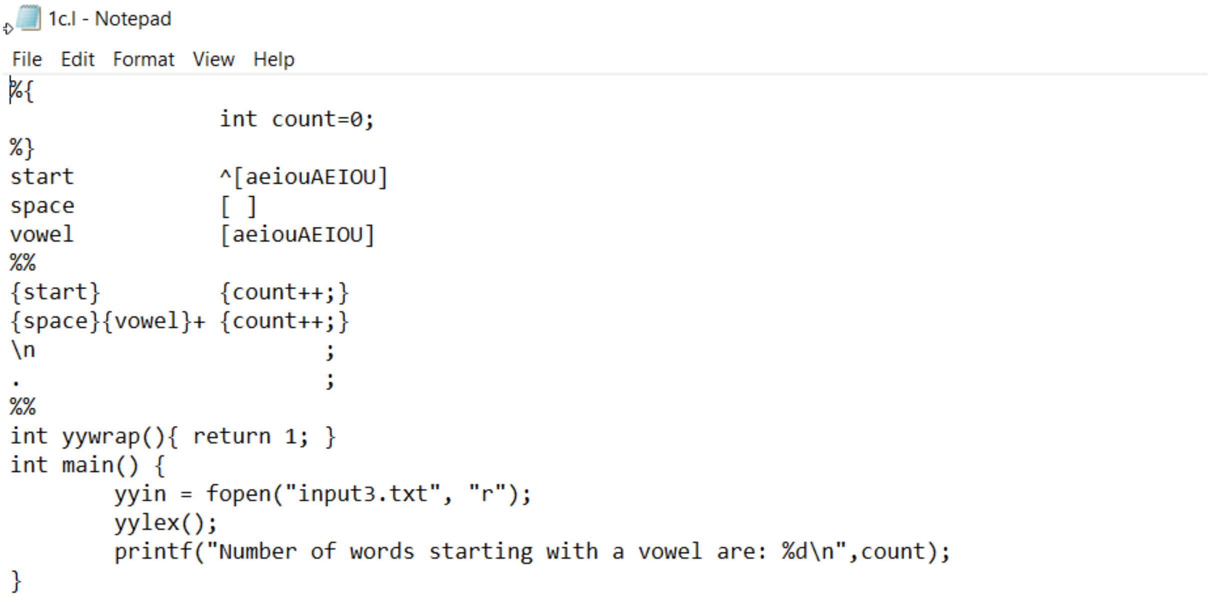
PROGRAM 3

Input Data



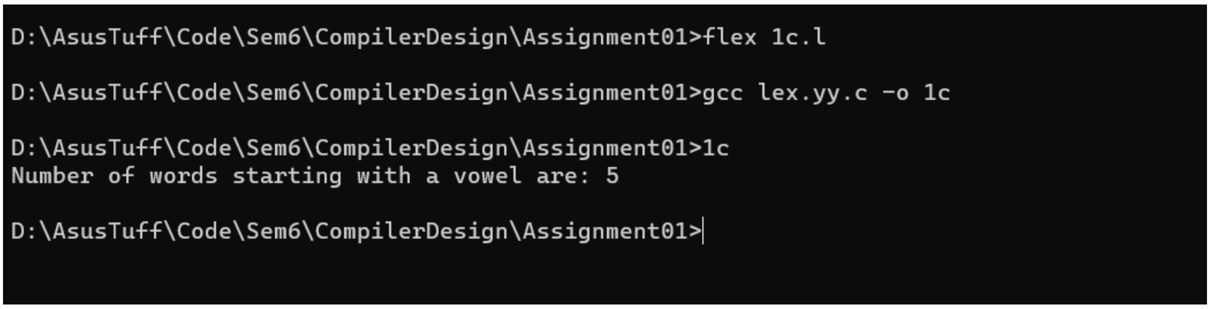
```
I am a good boy.
Uol study nn ju
let try this out
```

Lex File



```
%{
                int count=0;
%}
start           ^[aeiouAEIOU]
space           [ ]
vowel           [aeiouAEIOU]
%%
{start}         {count++;}
{space}{vowel}+ {count++;}
\n                      ;
.                       ;
%%
int yywrap(){ return 1; }
int main() {
        yyin = fopen("input3.txt", "r");
        yylex();
        printf("Number of words starting with a vowel are: %d\n",count);
}
```

Output



```
D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>flex 1c.l

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>gcc lex.yy.c -o 1c

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>1c
Number of words starting with a vowel are: 5

D:\AsusTuff\Code\Sem6\CompilerDesign\Assignment01>
```