ADITI BHATIA

D12A 06

SPCC LAB 03

Checking if number is odd or even:-

C: > Users > Student > Desktop > lexical analysis > even_odd.l

```
%{
#include<stdio.h>
int i;
%}

%%

[0-9]+     {i=atoi(yytext);
           if(i%2==0)
               printf("Even");
           else
          printf("Odd");}
%%

int yywrap(){}

/* Driver code */
int main()
{

    yylex();
    return 0;
}
```

```
^C
C:\Users\Student\Desktop\lexical analysis>flex even_odd.l

C:\Users\Student\Desktop\lexical analysis>gcc lex.yy.c -o even_odd.exe
even_odd.l:23:2: warning: no newline at end of file

C:\Users\Student\Desktop\lexical analysis>even_odd.exe
23
Odd
45
Odd
22
Even
12
Even
```

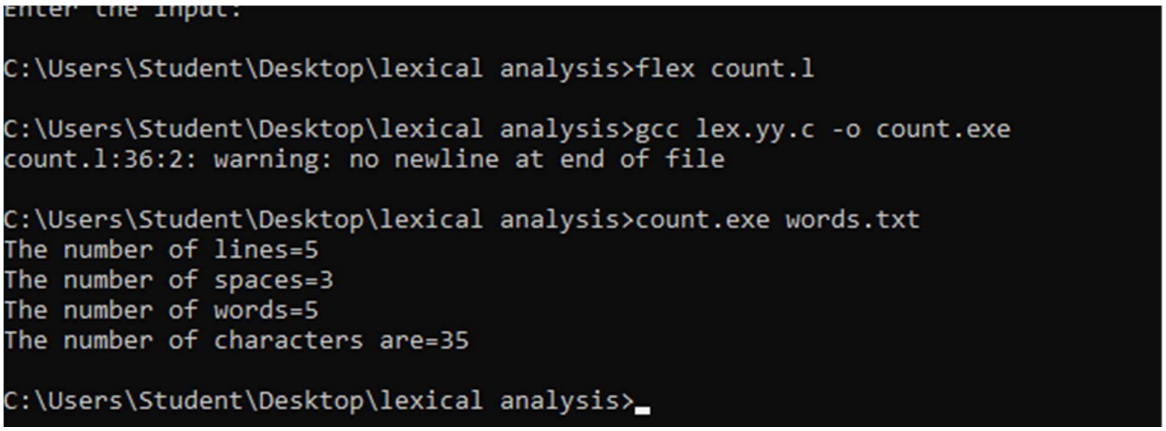
WORD Count:

C: > Users > Student > Desktop > lexical analysis > count.l

```
%{
#include<stdio.h>
int sc=0,wc=0,lc=0,cc=0;
%}

%%
[\n] { lc++; cc+=yyleng;}
[  \t] { sc++; cc+=yyleng;}
[^\t\n ]+ { wc++;  cc+=yyleng;}
%%

int main(int argc ,char* argv[ ])
{
    if (argc > 1)
{
 FILE *file;
 file = fopen(argv[1], "r");
 if (!file)
 {
 fprintf(stderr, "Could not open %s\n", argv[1]);
 exit(1);
 }
 yyin = file;
}

    yylex();
    printf("The number of lines=%d\n",lc);
    printf("The number of spaces=%d\n",sc);
```

C: > Users > Student > Desktop > lexical analysis > count.l

```
 }
 yyin = file;
}

    yylex();
    printf("The number of lines=%d\n",lc);
    printf("The number of spaces=%d\n",sc);
    printf("The number of words=%d\n",wc);
    printf("The number of characters are=%d\n",cc);
}

int yywrap( )
{
    return 1;
}
```


Enter the input:
C:\Users\Student\Desktop\lexical analysis>flex count.l
C:\Users\Student\Desktop\lexical analysis>gcc lex.yy.c -o count.exe
count.l:36:2: warning: no newline at end of file
C:\Users\Student\Desktop\lexical analysis>count.exe words.txt
The number of lines=5
The number of spaces=3
The number of words=5
The number of characters are=35
C:\Users\Student\Desktop\lexical analysis>

Printf Scanf:-

C: > Users > Student > Downloads > SPCC > countPrintStatements.l

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

%%
"printf" {fprintf(yyout,"writef"); pc++;}
"scanf" {fprintf(yyout,"readf"); sfc++;}
%%

main(int argc ,char *argv[])
{
    if(argc!=3){
        printf("Usage ./a.out in.txt out.txt\n");
        exit(0);
    }

    yyin=fopen(argv[1],"r");
    yyout=fopen(argv[2],"w");
    yylex();
    printf("\n the number of printf lines = %d\n",pc);
    printf("\n the number of scanf lines = %d\n",sfc);

}

int yywrap(){
    return 1;
}
```

```
C:\Users\Student\Downloads\SPCC>flex countPrintStatements.l

C:\Users\Student\Downloads\SPCC>gcc lex.yy.c -o countPrintStatements.exe
countPrintStatements.l:28:2: warning: no newline at end of file

C:\Users\Student\Downloads\SPCC>./countPrintStatements.exe code.txt empty.txt
'.' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Student\Downloads\SPCC>.\countPrintStatements.exe code.txt empty.txt

 the number of printf lines = 2

 the number of scanf lines = 1
```

English Grammar

C: > Users > Student > Downloads > SPCC > classifyEnglishGrammar.l

```
%}

%x verb_token
%x adverb_token
%x adjective_token
%x noun_token

%%

"run"|"walk"|"jump"|"swim"|"fly"|"eat"|"drink"|"sleep"|"play"|"work" {
    verbCount++;
    BEGIN(verb_token);
}
"quickly"|"slowly"|"happily"|"angrily"|"carefully"|"loudly"|"softly"|"well"|"badly"|"hard" {
    adverbCount++;
    BEGIN(adverb_token);
}

"happy"|"sad"|"angry"|"excited"|"tired"|"strong"|"weak"|"big"|"small"|"fast" {
    adjectiveCount++;
    BEGIN(adjective_token);
}
"dog"|"cat"|"bird"|"car"|"house"|"tree"|"flower"|"book"|"pencil"|"computer" {
    nounCount++;
    BEGIN(noun_token);
}
<verb_token>.|\n {BEGIN(INITIAL);}
<adverb_token>.|\n {BEGIN(INITIAL);}
<adjective_token>.|\n {BEGIN(INITIAL);}
<noun_token>.|\n {BEGIN(INITIAL);}
```

C: > Users > Student > Downloads > SPCC > classifyEnglishGrammar.l

```
%x noun_token

%%

"run"|"walk"|"jump"|"swim"|"fly"|"eat"|"drink"|"sleep"|"play"|"work" {
    verbCount++;
    BEGIN(verb_token);
}
"quickly"|"slowly"|"happily"|"angrily"|"carefully"|"loudly"|"softly"|"well"|"badly"|"hard" {
    adverbCount++;
    BEGIN(adverb_token);
}

"happy"|"sad"|"angry"|"excited"|"tired"|"strong"|"weak"|"big"|"small"|"fast" {
    adjectiveCount++;
    BEGIN(adjective_token);
}
"dog"|"cat"|"bird"|"car"|"house"|"tree"|"flower"|"book"|"pencil"|"computer" {
    nounCount++;
    BEGIN(noun_token);
}
<verb_token>.|\n {BEGIN(INITIAL);}
<adverb_token>.|\n {BEGIN(INITIAL);}
<adjective_token>.|\n {BEGIN(INITIAL);}
<noun_token>.|\n {BEGIN(INITIAL);}

. { /* ignore all other characters */ }

%%
```

C: > Users > Student > Downloads > SPCC > classifyEnglishGrammar.l

```
31  }
32  "dog"|"cat"|"bird"|"car"|"house"|"tree"|"flower"|"book"|"pencil"|"computer" {
33      nounCount++;
34      BEGIN(noun_token);
35  }
36  <verb_token>.|\n {BEGIN(INITIAL);}
37  <adverb_token>.|\n {BEGIN(INITIAL);}
38  <adjective_token>.|\n {BEGIN(INITIAL);}
39  <noun_token>.|\n {BEGIN(INITIAL);}
40
41  . { /* ignore all other characters */ }
42
43  %%
44
45  int yywrap()
46  {
47      return 1;
48  }
49
50  int main(int argc ,char *argv[])
51  {
52      yylex();
53      printf("\n the number of noun lines = %d\n",nounCount);
54      printf("\n the number of adverb lines = %d\n",adverbCount);
55      printf("\n the number of adjective lines = %d\n",adjectiveCount);
56      printf("\n the number of verb lines = %d\n",verbCount);
57       return 0;
58  }
59
```

```
C:\Users\Student\Downloads\SPCC>classifyEnglishGrammar.exe
the strong wind blew loudly in the trees

the cat jump out of the window


 the number of noun lines = 2

 the number of adverb lines = 1

 the number of adjective lines = 1
^C
```