**D12B\_01\_SANJANA ASRANI\_SPCC\_EXP-03\_FLEX**

1. COUNT NUMBER OF ***VOWELS & CONSONANTS***
   1. Input as a file

Code :

count\_vowel.l

%{

#include<stdio.h>

int c=0,v=0,p=0,s=0,op=0,i=0;

%}

%%

printf p++;

scanf s++;

[aeiouAEIOU] v++;

[a-zA-Z] c++;

[\+\-\*/] op++;

([a-zA-Z0-9])\* i++;

%%

int yywrap() {

return 1;

}

int main()

{

yyin = fopen("count\_vowel.txt", "r");

yylex();

printf("Vowels = %d & Consonants = %d\n",v,c);

printf("Number of printf = %d and number of scanf = %d\n",p,s);

printf("Number of operators = %d\n",op);

printf("Number of variables = %d\n",i);

return 0;

}

count\_vowel.txt

a e i z o s

printf

scanf

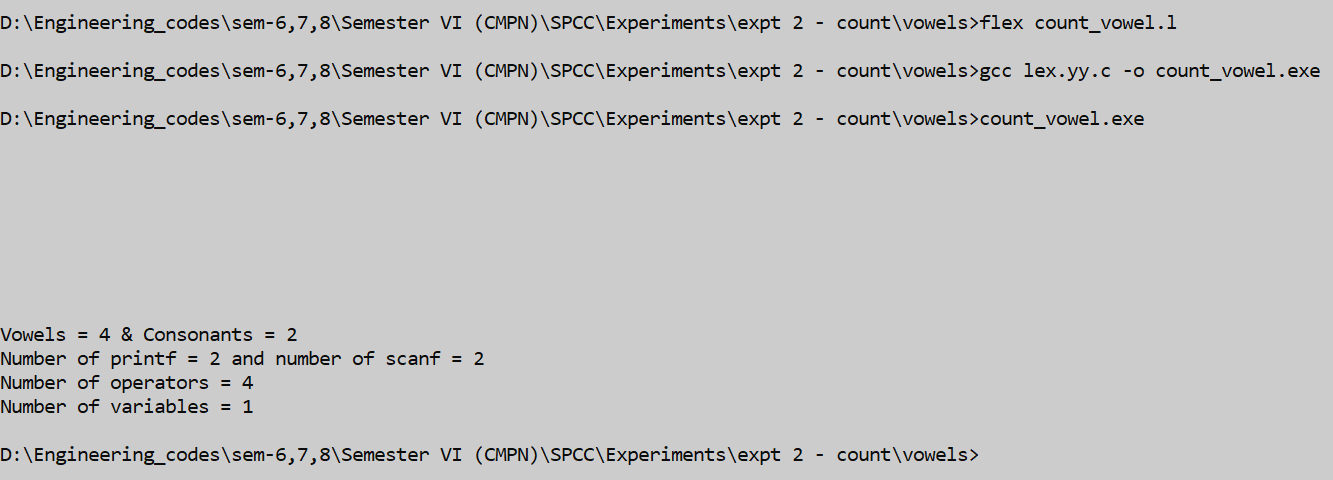
+-\*/

printf

scanf

xyz

Output :



* 1. Cli input

Code :

vc.l

%{

#include<stdio.h>

int v=0,c=0;

%}

%%

[aeiouAEIOU]              {v++;}

[a-zA-Z]                  {c++;}

.                        { }

%%

int yywrap(void) {

return 1;

}

int main()

{

printf("Enter the Text:" );

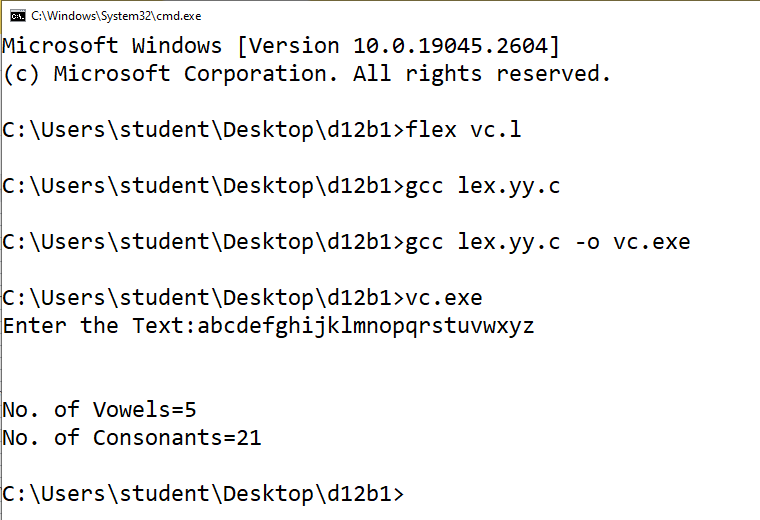
yylex();

printf("\nNo. of Vowels=%d\nNo. of Consonants=%d\n",v,c);

return 0;

}

Output :



1. CHECK IF A NUMBER IS ***EVEN OR ODD***
   1. Input from file :

Code :

even\_odd.l

%{

#include<stdio.h>

int no\_of\_odds = 0;

int no\_of\_evens = 0;

int i=0;

%}

%%

[0-9]+ {i = atoi(yytext);

if(i%2!=0){

no\_of\_odds++;

printf("%d is odd",i);

}

else{

printf("%d is even",i);

no\_of\_evens++;

}

}

%%

int yywrap(){}

int main(){

yyin = fopen("input.txt","r");

yylex();

printf("\nThe number of odds %d\n",no\_of\_odds);

printf("The number of evens %d\n",no\_of\_evens);

return 0;

}

input.txt

1

2

3

4

5

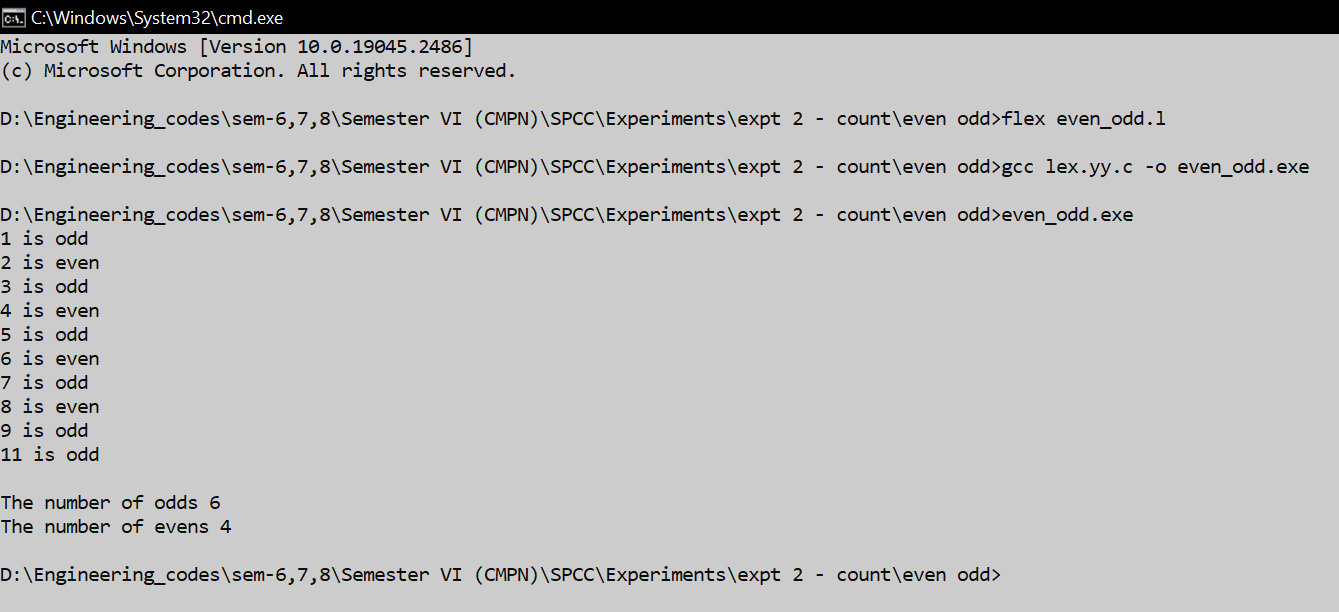
6

7

8

9

11

Output :

* 1. Input from cli

Code :

evenOdd.l

%{

    #include<stdio.h>

    int num;

    int flag;  // flag=0 means even

%}

%%

[0-9]+     {num=atoi(yytext);

           if(num%2==0) { flag=0;}

           else { flag=0;}

       }

%%

int yywrap() {

    return 1;

}

int main() {

printf("enter an integer: ");

yylex();

if (flag==0) {

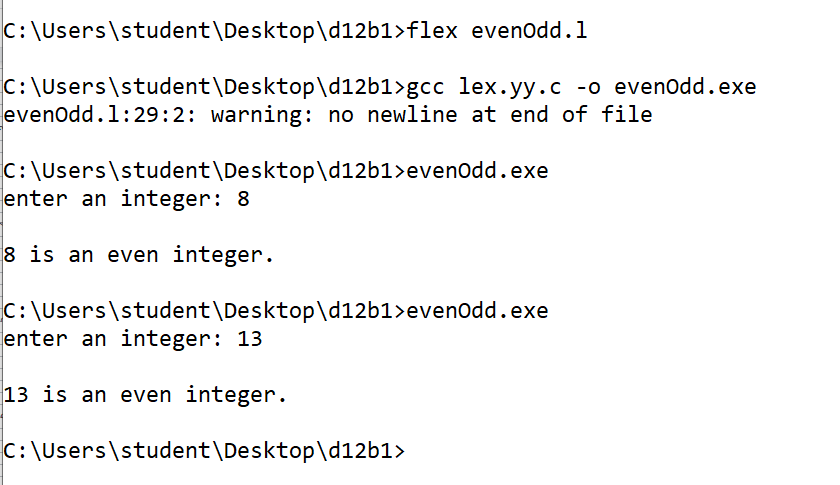
printf("%d is an even integer.\n",num); }

else printf("%d is an even integer.\n",num);

return 0;

}

Output :



1. COUNT NUMBER OF **PRINTF & SCANF** STATEMENTS

Code :

count\_printscan.l

%{

#include<stdio.h>

int p=0,s=0;

%}

%%

printf p++;

scanf s++;

%%

int yywrap() {

return 1;

}

int main()

{

yyin = fopen("randomFile.c", "r");

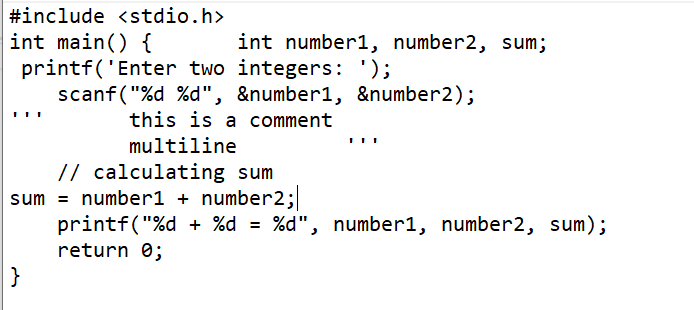
yylex();

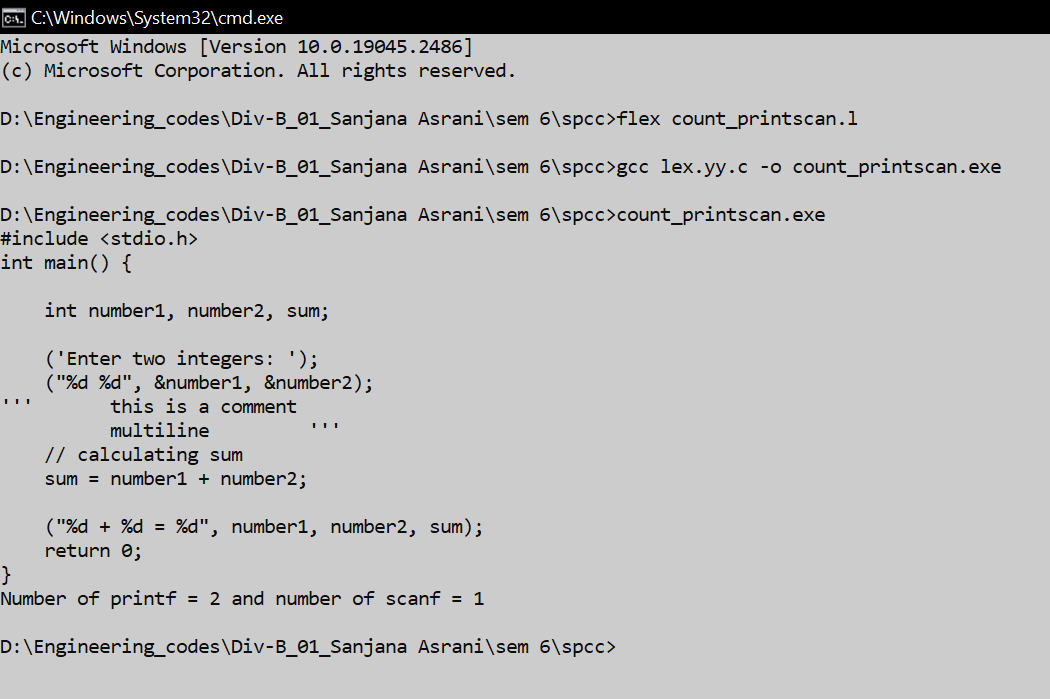
printf("Number of printf = %d and number of scanf = %d\n",p,s);

return 0;

}

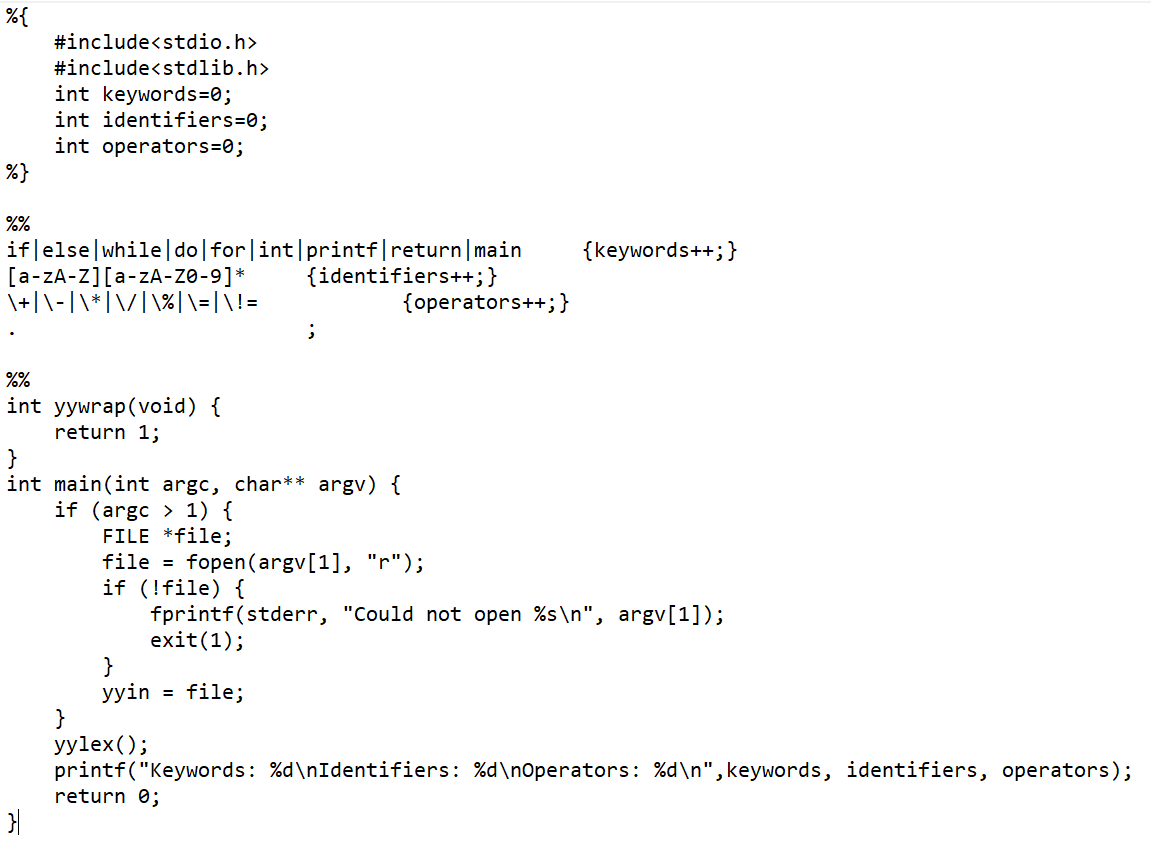
randomFile.c

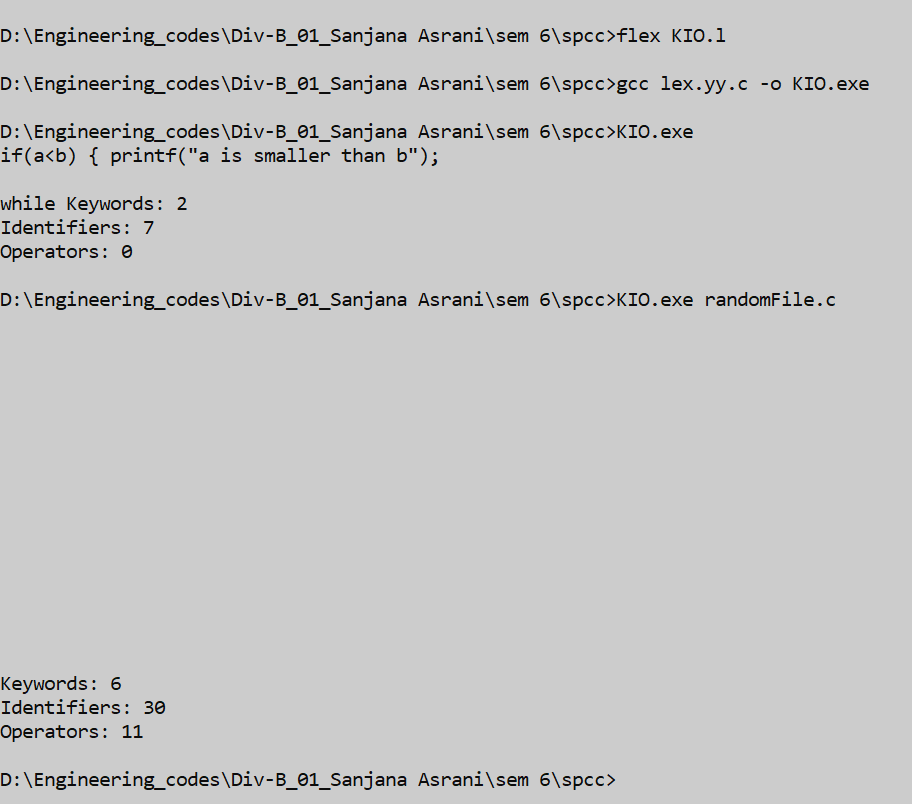
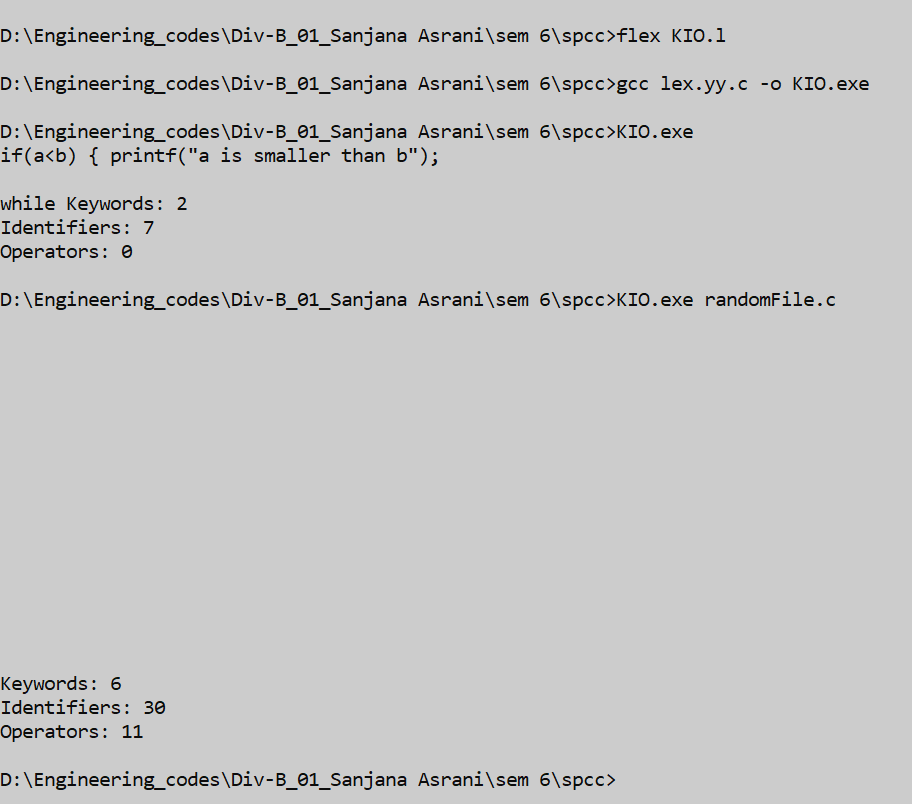
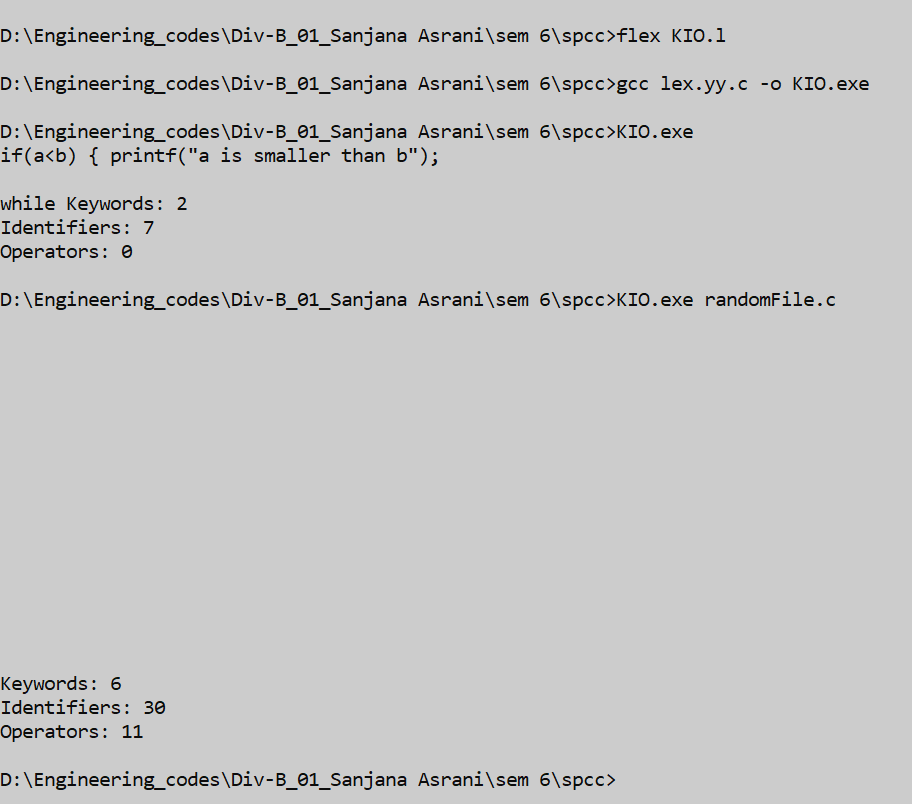


Output: 

1. COUNT NUMBER OF **OPERATORS, IDENTIFIERS, KEYWORDS** AND LIST THEM

Code : KIO.l



Output :

5. CLASSIFY **ADVERBS, VERBS**

5.1. using cli input

Code :

count\_verb.l

%{

#include <stdio.h>

#include <string.h>

int verb\_count = 0;

int adverb\_count = 0;

int adjective\_count = 0;

int noun\_count = 0;

%}

%x verb\_token

%x adverb\_token

%x adjective\_token

%x noun\_token

%%

"run"|"walk"|"jump"|"swim"|"fly"|"eat"|"drink"|"sleep"|"play"|"work" {

verb\_count++;

BEGIN(verb\_token);

}

"quickly"|"slowly"|"happily"|"angrily"|"carefully"|"loudly"|"softly"|"well"|"badly"|"hard" {

adverb\_count++;

BEGIN(adverb\_token);

}

"happy"|"sad"|"angry"|"excited"|"tired"|"strong"|"weak"|"big"|"small"|"fast" {

adjective\_count++;

BEGIN(adjective\_token);

}

"dog"|"cat"|"bird"|"car"|"house"|"tree"|"flower"|"book"|"pencil"|"computer" {

noun\_count++;

BEGIN(noun\_token);

}

<verb\_token>.|\n { BEGIN(INITIAL); }

<adverb\_token>.|\n { BEGIN(INITIAL); }

<adjective\_token>.|\n { BEGIN(INITIAL); }

<noun\_token>.|\n { BEGIN(INITIAL); }

%%

int yywrap(){

return 1;

}

int main(int argc, char \*argv[])

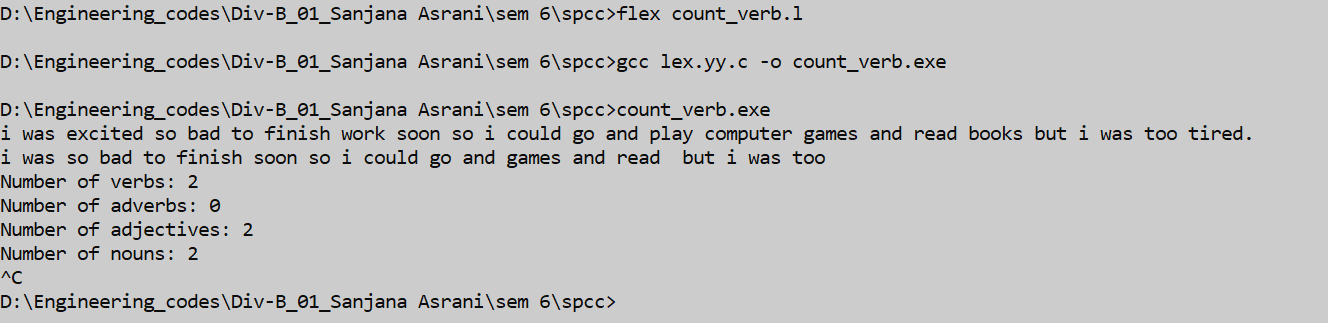
{

yylex();

printf("Number of verbs: %d\nNumber of adverbs: %d\nNumber of adjectives: %d\nNumber of nouns: %d\n", verb\_count, adverb\_count,adjective\_count,noun\_count);

return 0;

}

Output :

5.2 taking file as input

Code :

verbAdverb.l

%{

#include <stdio.h>

#include <string.h>

int verb\_count = 0;

int adverb\_count = 0;

%}

%x verb\_token

%x adverb\_token

%%

"run"|"walk"|"jump"|"swim"|"fly"|"eat"|"drink"|"sleep"|"play"|"work" {

verb\_count++;

BEGIN(verb\_token);

}

"quickly"|"slowly"|"happily"|"angrily"|"carefully"|"loudly"|"softly"|"well"|"badly"|"hard" {

adverb\_count++;

BEGIN(adverb\_token);

}

<verb\_token>.|\n { BEGIN(INITIAL); }

<adverb\_token>.|\n { BEGIN(INITIAL); }

%%

int yywrap(){

return 1;

}

int main()

{

yyin = fopen("temp\_file.txt", "r");

yylex();

printf("Number of verbs: %d\nNumber of adverbs: %d\n", verb\_count, adverb\_count);

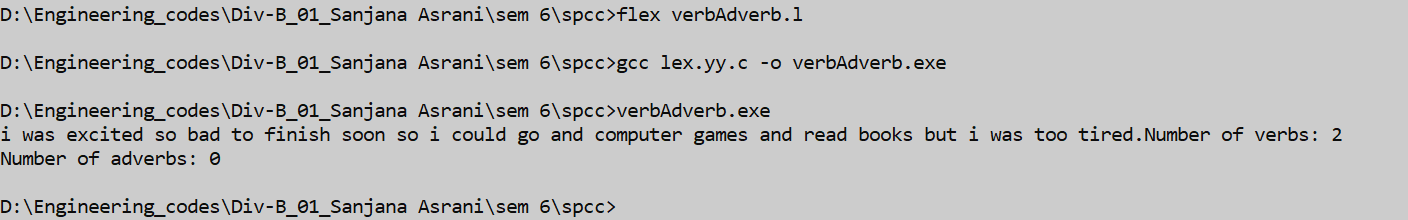
return 0;

}

temp\_file.txt

i was excited so bad to finish work soon so i could go and play computer games and read books but i was too tired.

Output :



6.COUNT NO. OF **LINES, CHARACTERS, WORDS**

6.1. Input from file :

Code :

count.l

%{

#include<stdio.h>

int nlines=0, nc=0, nword=0;

%}

%%

" " {nword++;} //word count

\n {nlines++;nword++;} //line counter

. {nc++;} //character counter

end {return 0;}

%%

int yywrap(){return 1;}

int main(){

yyin=fopen("sample.txt","r");

yylex();

printf("\n number of lines is %d",nlines);

printf("\n number of Characters is %d",nc);

printf("\n number of word is %d\n",nword);

return 0;

}

sample.txt

Hello world

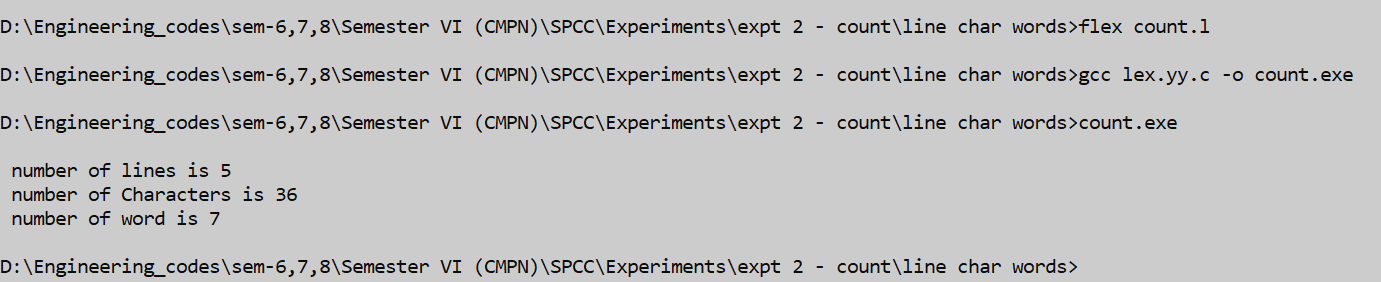
sanjana

d12b1

asrani

sem6 spcc

Output :



6.2. user input from cli

Code :

LWC.l

%{

#include <stdio.h>

#include <string.h>

int word\_count = 0, char\_count = 0, line\_count = 0;

%}

%%

[A-Za-z]+ { word\_count++; char\_count += strlen(yytext); }

. { char\_count++; }

\n { line\_count++; }

%%

int yywrap() {

return 1;

}

int main() {

yylex();

printf("Word count: %d\nCharacter count: %d\nLine count: %d\n ", word\_count, char\_count, line\_count);

return 0;

}

Output :

