U18CO018

Shubham Shekhaliya

Assignment – 6 (SS)

1-> Write a Lex program to count the number of lines, characters and words of the given input file.

Code :-

%{

#include<stdio.h>

int lines=0, words=0,total=0;

%}

%%

\n { lines++; words++;}

[\t ' '] words++;

. total++;

%%

int main(void)

{

    char s[100];

    printf( "Enter a file name :");

   gets( s );

yyin= fopen(s,"r");

yylex();

printf(" This File contains ...");

printf("\n\t%d lines", lines);

printf("\n\t%d words",words);

printf("\n\t%d characters.\n",total);

}

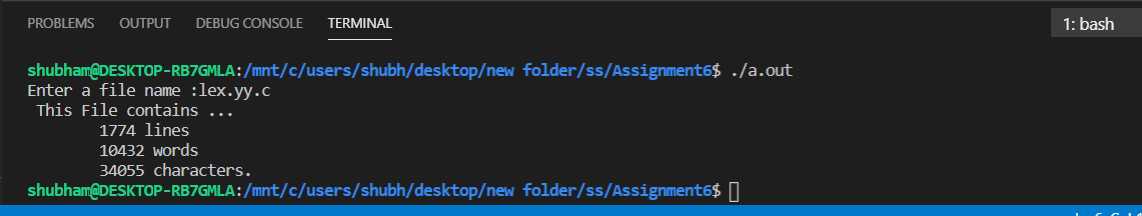
int yywrap()

{

return(1);

}

Output:-



2-> Design a scanner to

(a) Count number of single and multiple line comments from a C program available in xyz.txt file. [Note: You can create any txt file having sample C code which contains single and multiple line comments]

(b) Remove comment lines from the C program.

Code:-

%{#include<stdio.h>

int count1 = 0, count2;

%}

%x C

%%

"/\*"][.]\*"\*/"   {count2++;}

"/\*"        {BEGIN C;}

<C>"\*/"     {BEGIN 0; count2++;}

<C>\n       {;}

<C>.        {;}

\/\/.\*      {count1++;}

%%

void main() {

    char file[] = "data.c";

    yyin = fopen(file , "r");

    yylex();

    printf("\nNumber of comment in c file %s\nSingle line : %d\nMultiple line : %d\n", file, count1, count2);

}

int yywrap() {

    return 1;

}

Input :-

#include <stdio.h>

//Testing comment line 1

/\* Testing comment

line 2 \*/

void main()

{

    //Testing comment line 3

    printf("Hello world!\n");

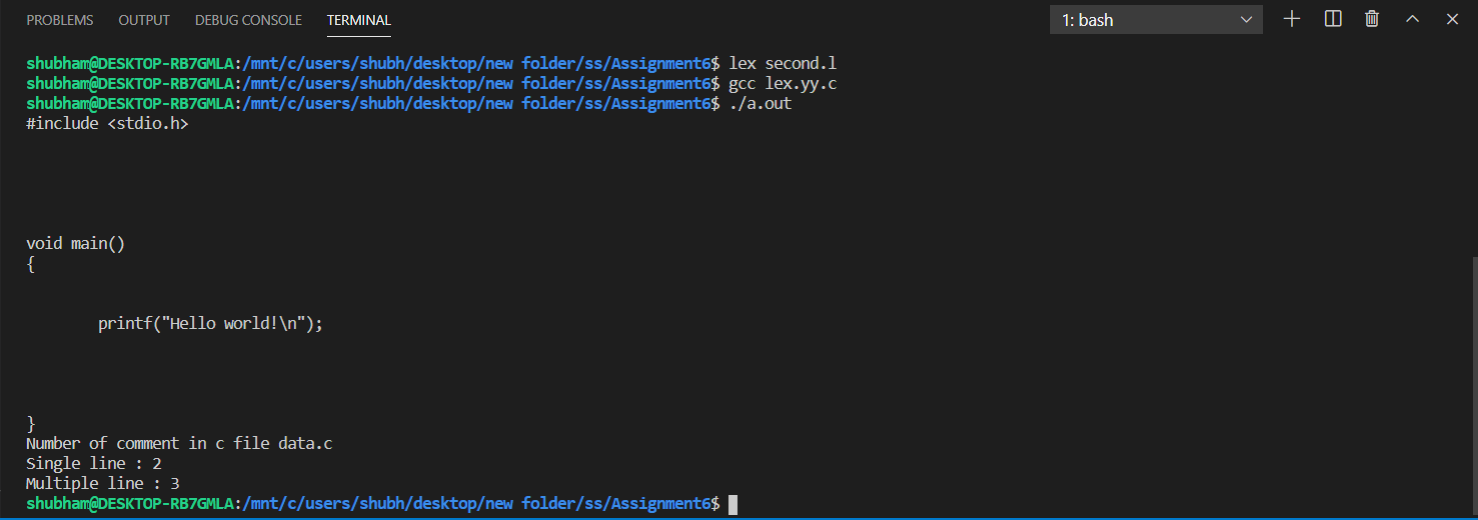
    /\* Testing 4 //

    Comment Line \*/

    /\* Testing comment line 5 \*/

}

Output:-



3-> Write a Lex program to check valid/invalid

1. Mobile number (considering 10-digit mobile number followed by country code +91)

Code:-

%{#include<stdio.h>

    /\* Definition section \*/

%}

/\* Rule Section \*/

%%

[+][9][1][1-9][0-9]{9} {printf("\nMobile Number Valid\n");}

.+ {printf("\nMobile Number Invalid\n");}

%%

// driver code

int main()

{

    printf("\nEnter Mobile Number : ");

    yylex();

    printf("\n");

}

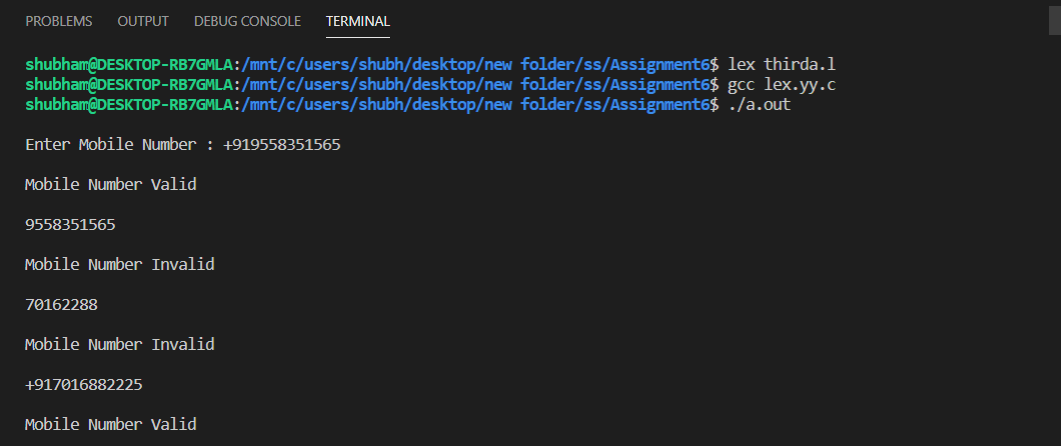
int yywrap()

{

    return 1;

}

Output:-



(b) Email address

Code :-

%{

#include<stdio.h>

%}

%%

^[a-z][a-z0-9\_]\*(@[A-Za-z]+)(\.[a-z]+)+ {printf("valid");}

.\* {printf("invalid");}

%%

int main()

{

yylex();

}

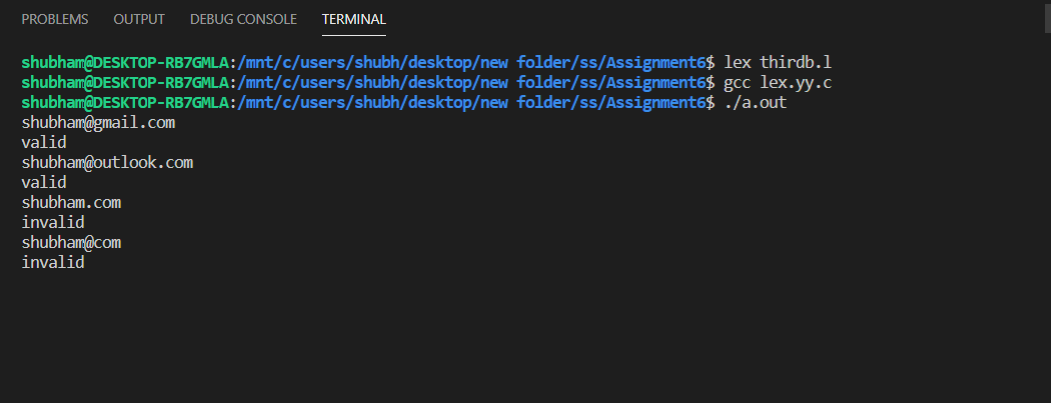
int yywrap()

{

    return 1;

}

Output :-



4. Design a scanner to check whether a number is Armstrong number or not.

Code :-

%{

#include<string.h>

#include<math.h>

void check(char \*);

%}

%%

[0-9]+ check(yytext);

%%

int main()

{

    extern FILE \*yyin;

    yyin=fopen("num","r");

    yylex();

}

void check(char \*a)

{

    int len=strlen(a),i,num=0;

    for(i=0;i<len;i++)

        num=num\*10+(a[i]-'0');

    int x=0,temp=num;

    while(num>0)

    {

        int tt = 1;

        int dd = num%10;

        for(int i = 0; i<len;i++) {

            tt \*= dd;

        }

        x=x+tt;

        num=num/10;

    }

    if(x==temp)

        printf("%d is armstrong \n",temp);

    else

        printf("%d is not armstrong \n",temp);

}

int yywrap()

{

    return 1;

}

Output:-

