#include<stdio.h>

#include<string.h>

int main()

{

FILE \*fp;

char ch,c;

int i=0,j,flag=0,to=0;

char s[100][100]={"do","double","else","enum","extern","float","for","goto","if","int","long","register","return","short","signed","sizeof","static","struct",

"switch","typedef","union","unsigned","void","volatile","while","continue","break","case","char","const","default","auto"};

char st[10];

fp=fopen("file.txt","r");

printf("Token no\t\tToken\t\tLexame \n");

while((ch=getc(fp))!=EOF)

{

i=0;

flag=0;

if(ch>=42&&ch<=47)

{

to++;

printf("%d\t\toperator\t\t%c\n",to,ch);

}

if((ch>=32&&ch<=41)||(ch>=91&&ch<=96)||(ch>=58&&ch<=63)||(ch>=123&&ch<=126))

{

to++;

printf("%d\t\tspecial character\t\t%c\n",to,ch);

}

if(isdigit(ch))

{

to++;

printf("%d\t\tdigit\t\t\t%c\n",to,ch);

}

if(isalpha(ch))

{

st[i]=ch;

i++;

ch=getc(fp);

while(isalnum(ch)&&ch!=' ')

{

st[i]=ch;

i++;

ch=getc(fp);

}

st[i]='\0';

for(j=0;j<32;j++)

{

if(strcmp(st,s[j])==0)

{

flag=1;

break;

}

}to++;

if(flag==1)

{

printf("%d\t\tkeyword\t\t\t%s \n",to,st);

}

else

{

printf("%d\t\tidentifier\t\t%s\n",to,st);

}

if(ch=='.')

{

i=0; int k=1;

if(isalpha(c=getc(fp)))

{

st[i]=c;

i++;

c=getc(fp);

while(isalnum(c)&&c!=' ')

{

st[i]=c;

i++;

k++;

c=getc(fp);

}

st[i]='\0';

to++;

printf("%d\t\tExtension\t\t.%s\n",to,st);

ch=c;

}

else

{

if(ch>=42&&ch<=47)

{

to++;

printf("%\t\toperator\t\t%c\n",to,ch);

}

if((c>=32&&c<=41)||(c>=91&&c<=96)||(c>=58&&c<=63)||(c>=123&&c<=126))

{to++;

printf("%d\t\tspecial character\t\t%c\n",to,c);

}

}

}

if((ch>=32&&ch<=41)||(ch>=91&&ch<=96)||(ch>=58&&ch<=63)||(ch>=123&&ch<=126))

{to++;

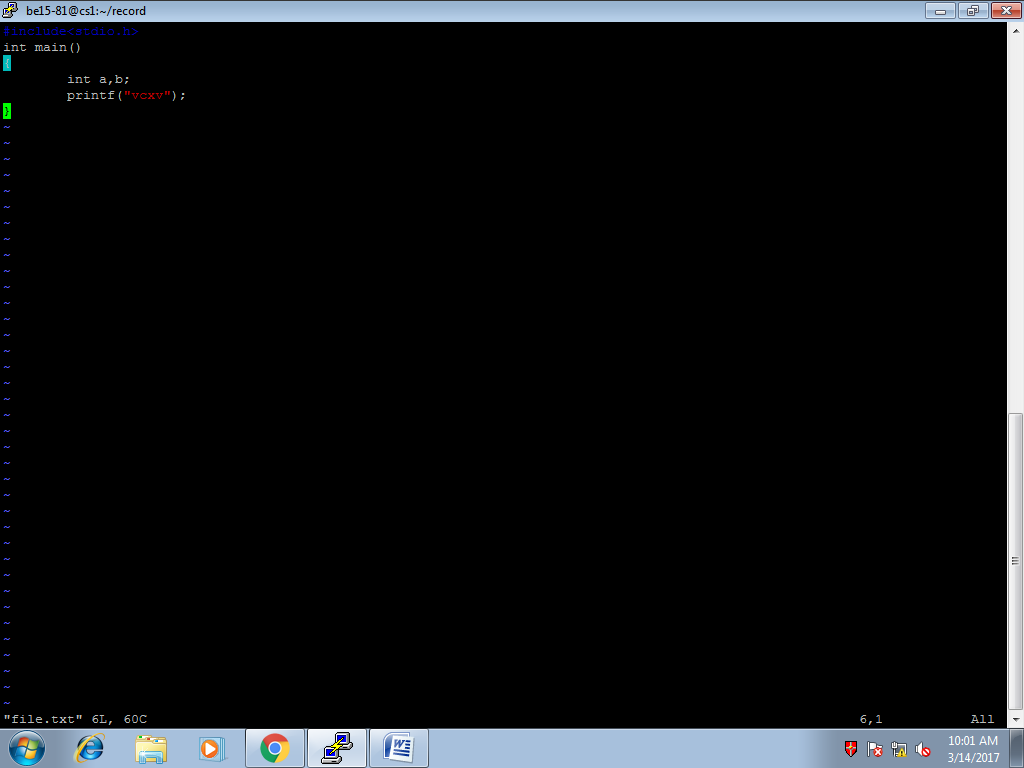
printf("%d\t\tspecial character\t\t%c\n",to,ch);

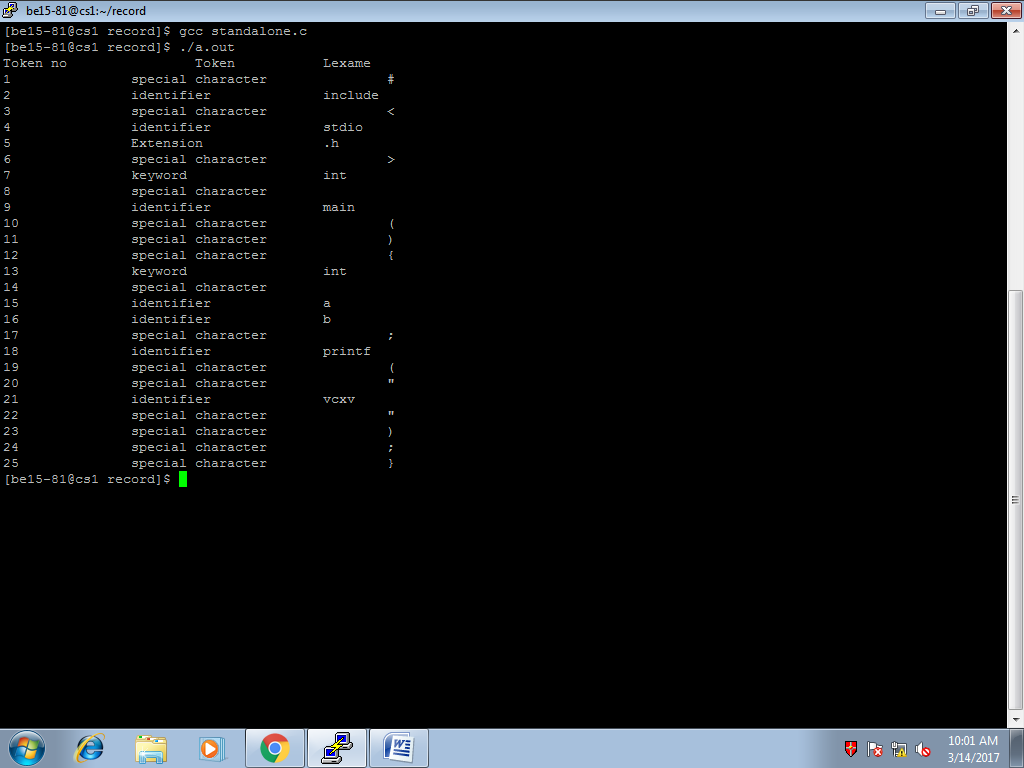
}

}

}

}





#include<stdio.h>

#include<math.h>

#include<stdlib.h>

static  int sumint;

 float sumfloat;

int main()

{

 FILE \*fp;

 char s[100];

 char c;

 int sum=0,ws;

 int i=0,j=0,lines=0,words=0,integers=0,floats=0,check;

 fp = fopen("file.txt","r");

 while((c=fgetc(fp))!=EOF)

        {

        if(c=='\n')

                        lines++;

                if(c!=' '&&c!='\n')

            {

                        s[i]=c;

                        i++;

                }

                if(c==' '||c=='\n')

                {

                        words++;

                        ws=i;

                        check=verify(s,i);

                        if(check==0)

                                integers++;

                         else if(check==2)

                                floats++;

                        i=0;

                }

        }

        printf("\nNumber of words:%d\n",words);

        printf("Number of lines:%d\n",lines);

        printf("Number of integers:%d\n",integers);

        printf("Number of floats:%d\n",floats);

        printf("Sum of integers:%d\n",sumint);

        printf("Sum of floats:%g\n",sumfloat);

}

int verify(char \*c,int x)

{

        int i=0,flag=0,temp=0;

        float temp1=0.0;

        char a[100],a1[100];

         for(i=0;i<x;i++)

        {

                if(((c[i])>=48&&(c[i])<=57)&&(c[i]!='+'||c[i]!='-'))

                        flag++;

        }

        i=0;

        if(c[i]=='+'||c[i]=='-')

        {

                flag=1;

                for(i=0;i<x;i++)

                {

                        if(((c[i])>=48&&(c[i])<=57)&&(c[i]!='+'||c[i]!='-'))

                                flag++;

    }

        }

        if(flag==x)

        {

                i=0;

                if(c[0]=='+'||c[0]=='-')

                {

                        for(i=1;i<x;i++)

{

  a[i-1]=c[i];

}

a[i-1]='\0';

                        temp=atoi(a);

                        if(c[0]=='+')

                        {

     sumint=sumint+temp;

                     }

                        else if(c[0]=='-')

                        {

       sumint=sumint-temp;

}

                }

                else

                {

                        for(i=0;i<x;i++)

{

 a[i]=c[i];

}

a[i]='\0';

                        temp=atoi(a);

                        sumint=sumint+temp;

                }

                return 0;

        }

         else

        {

                i=0;flag=0;int q=0;

                for(i=0;i<x;i++)

                {

                        if(((c[i])>=48&&(c[i])<=57&&(c[i]!='+'||c[i]!='-')))

                                flag++;

                        if(c[i]=='.')

                         {

       flag++;

       q++;

}

if(q>1)

break;

                }

                i=0;

                if(c[0]=='+'||c[0]=='-')

                {q=0;

                        flag=1;

                        for(i=0;i<x;i++)

                        {

                                if(((c[i])>=48&&(c[i])<=57&&(c[i]!='+'||c[i]!='-')))

                                        flag++;

                                          if(c[i]=='.')

                                       {

 flag++;

 q++;

}

if(q>1)

break;

                        }

                }

                if(flag==x)

                {

                        i=0;

                        if(c[0]=='+'||c[0]=='-')

                        {

                                for(i=1;i<x;i++)

                                        a1[i-1]=c[i];

                                        a1[i-1]='\0';

                                temp1=atof(a1);

                                if(c[0]=='+')

                                        sumfloat=sumfloat+temp1;

                                else if(c[0]=='-')

                                        sumfloat=sumfloat-temp1;

                        }

                        else

                        {

                         int p=0,m=0;

                         for(i=0;i<x;i++)

                         {

a1[i]=c[i];

}

                         a1[i]='\0';

                         for(i=0;a1[i]!='.';i++)

                         {

                         p++;

}

for(m=0;p>0;m++)

{

char sp=a1[m];

 sp=sp-'0';

temp1=temp1+(sp\*pow(10,--p));

}

p=0;

i++;

m++;

for(;a1[i]!='\0';i++)

{

p++;

}

for(;p>0;m++)

{

char np=a1[m];

np=np-'0';

temp1=temp1+(np\*pow(10,-p));

p--;

}

                                sumfloat=sumfloat+temp1;

                     }

                        return 2;

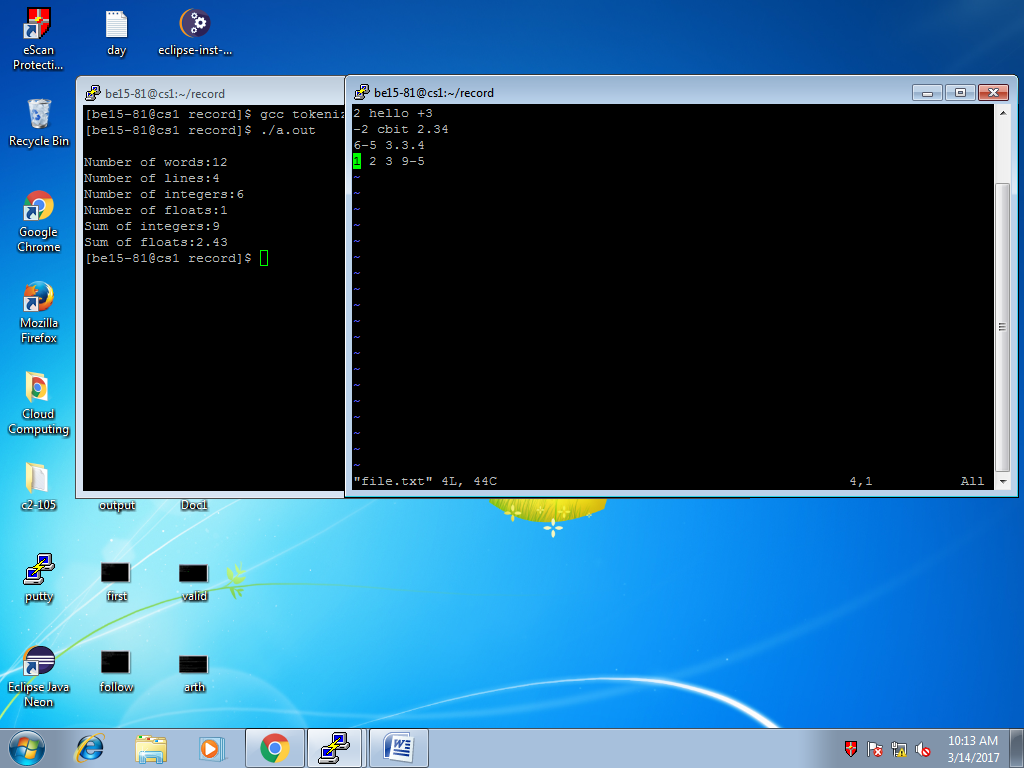
                }

                else

                        return -1;

        }

}



SCANNER

%{

#include<stdio.h>

#include<string.h>

int t=1;

int l=1;

%}

key auto|break|case|char|const|continue|default|do|double|else|enum|extern|float|for|goto|if|int|long|register|return|short|signed|sizeof|static|struct|switch|typedef|union|unsigned|void|volatile|while

file txt|h|c

%%

[#$&^{}()"'] {printf("\n%d\tspecial char\t%s",t,yytext);t++;}

{key} {printf("\n%d\tkeyword\t\t%s",t,yytext);t++;}

[a-z]+"."{file} {printf("\n%d\tfile\t\t%s",t,yytext);t++;}

[a-z]+"."[a-z]+ {printf("\n%d\tidentifier\t%s",t,yytext);t++;}

"++" {printf("\n%d\tincrementer\t%s",t,yytext);t++;}

"--" {printf("\n%d\tdecrementer\t%s",t,yytext);t++;}

"/\*"|"//" {printf("\n%d\tcomment begins\t%s",t,yytext);t++;}

"\*/" {printf("\n%d\tcomment ends\t%s",t,yytext);t++;}

[=+-/%<>] {printf("\n%d\toperator\t%s",t,yytext);t++;}

"\\"[btvrfn?0a] {printf("\n%d\tescape char\t%s",t,yytext);t++;}

[+-]?[0-9]+ {printf("\n%d\tdigit\t\t%s",t,yytext);t++;}

[a-zA-Z][a-zA-Z0-9\_]\* {printf("\n%d\tidentifier\t%s",t,yytext);t++;}

[;] {printf("\n%d\tterminator\t%s",t,yytext);t++;}

[\n] {l++;}

%%

int yywrap()

{

        return 1;

}

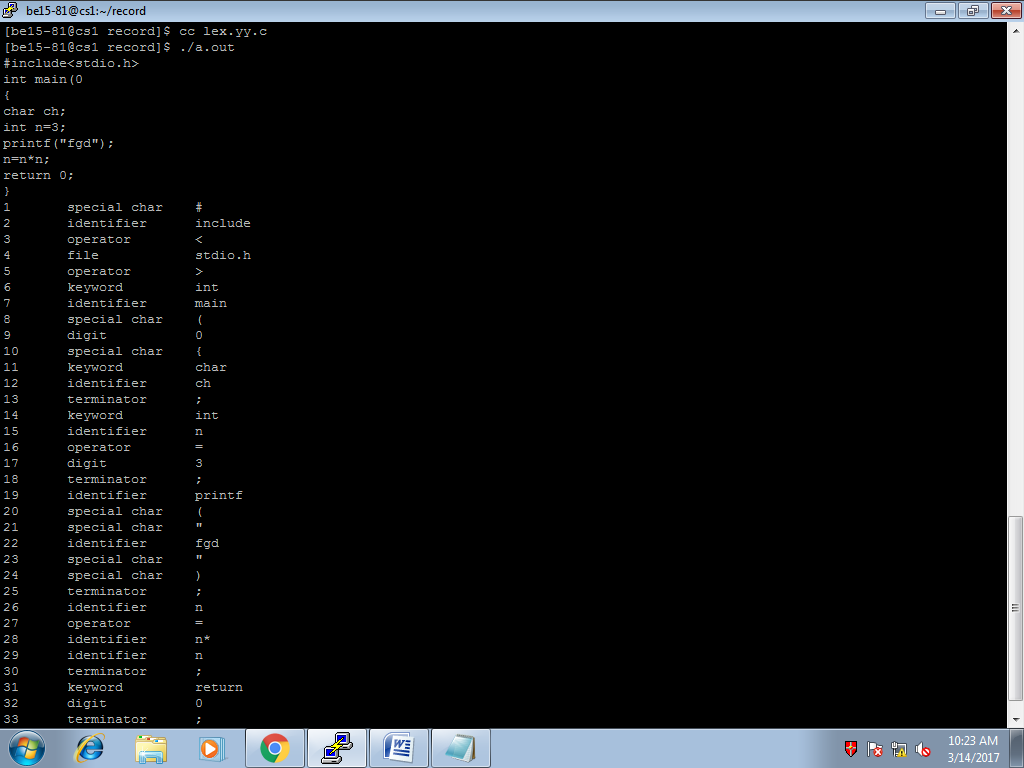
int main()

{

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

        yylex();

}



//hexaoctal

%{

#include<stdio.h>

%}

Hex [0][X|x][0-9a-fA-F]+

Oct [0][0-7]+

dec [+|-|0-9][0-9]+

%%

{Hex} printf("\n%s is Hexadecimal",yytext);

{Oct} printf("\n%s Octal",yytext);

{dec} printf("\n%s Decimal",yytext);

[\*] {return 1;}

%%

int main()

{

printf("\nEnter a string");

printf("\n \* to exit");

yylex();

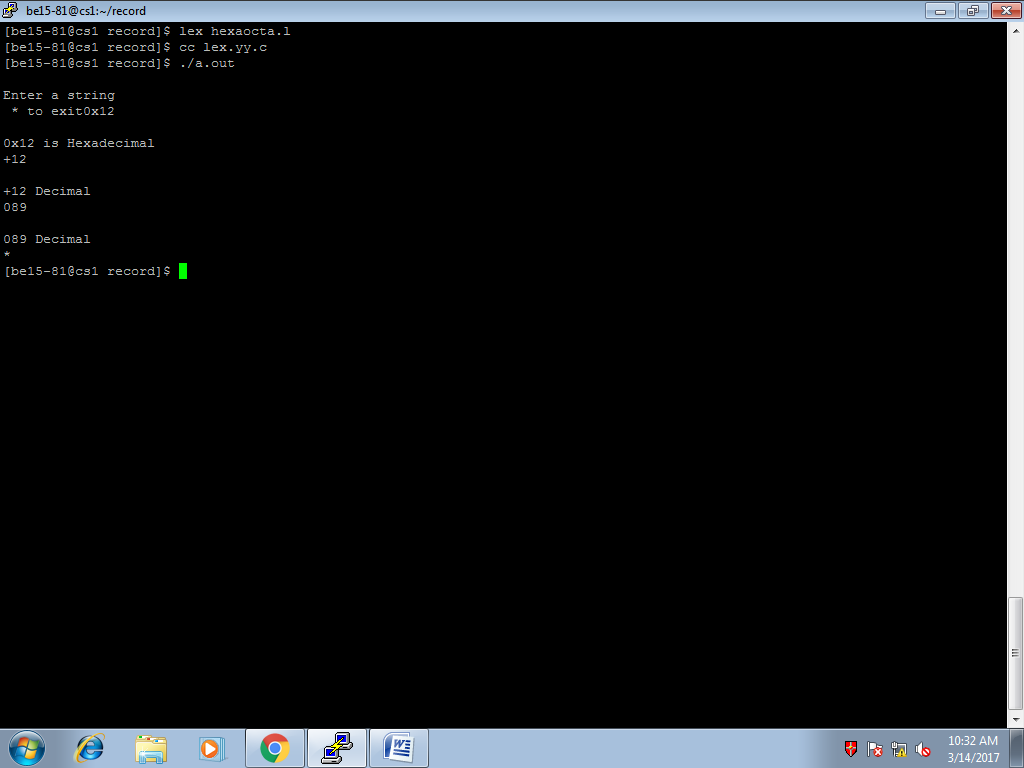
}

int yywrap()

{

return 1;

}



%{

#include<stdio.h>

%}

low [a-z]

upp [A-Z]

%%

{low} {printf("%c",yytext[0]-32);}

{upp} {printf("%c",yytext[0]+32);}

[\*] {return 1;}

%%

int main()

{

printf("Enter text to convert\n");

printf("\* to exit");

yylex();

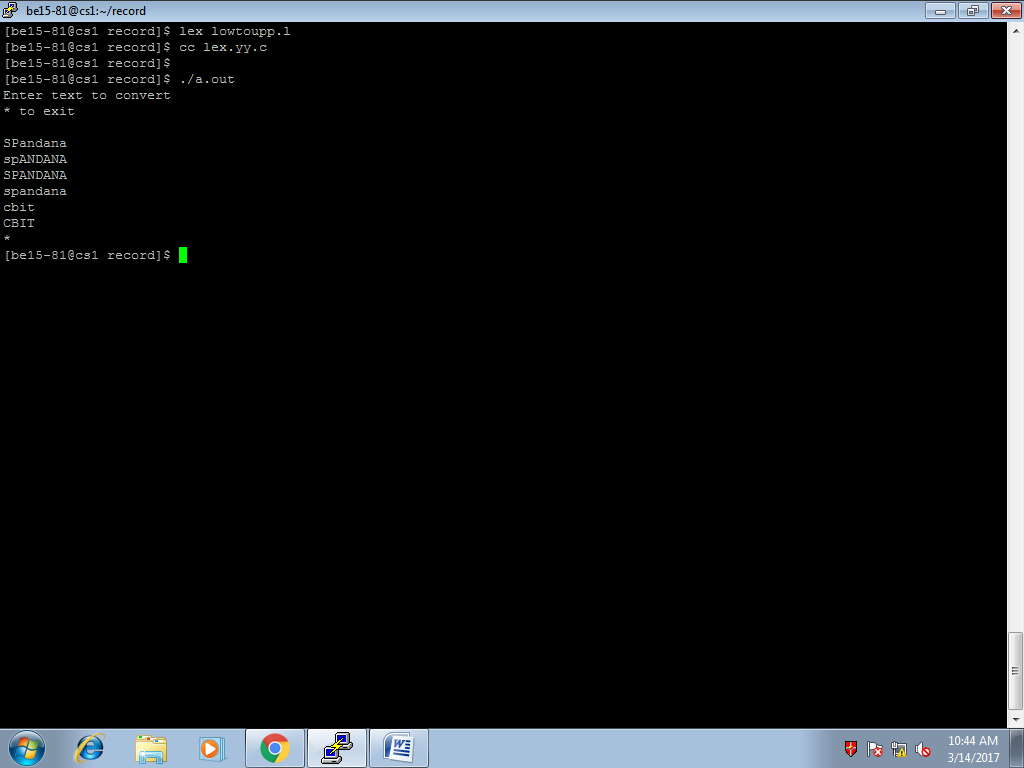
}

int yywrap()

{

return 1;

}



%{

#include<stdio.h>

#include<string.h>

int f,i,j;

%}

%%

[+-]?[0-9]+ {printf("\n%s is an integer!",yytext);}

[+-]?[0-9]\*[.][0-9]+ {f=0; for(i=0;i<yyleng;i++)

{if(yytext[i]=='.') { j=i+1; break;} }

 for(;j<yyleng;j++)

f++;

 printf("\n%s is a floating number with a precision of %d!",yytext,f);}

[0-9a-zA-Z]+[.][0-9+-.a-zA-Z]+ {printf("\ninvalid!!!");}

[\n]  {return 0;}

%%

int main()

{

printf("Enter a number :\n");

yylex();

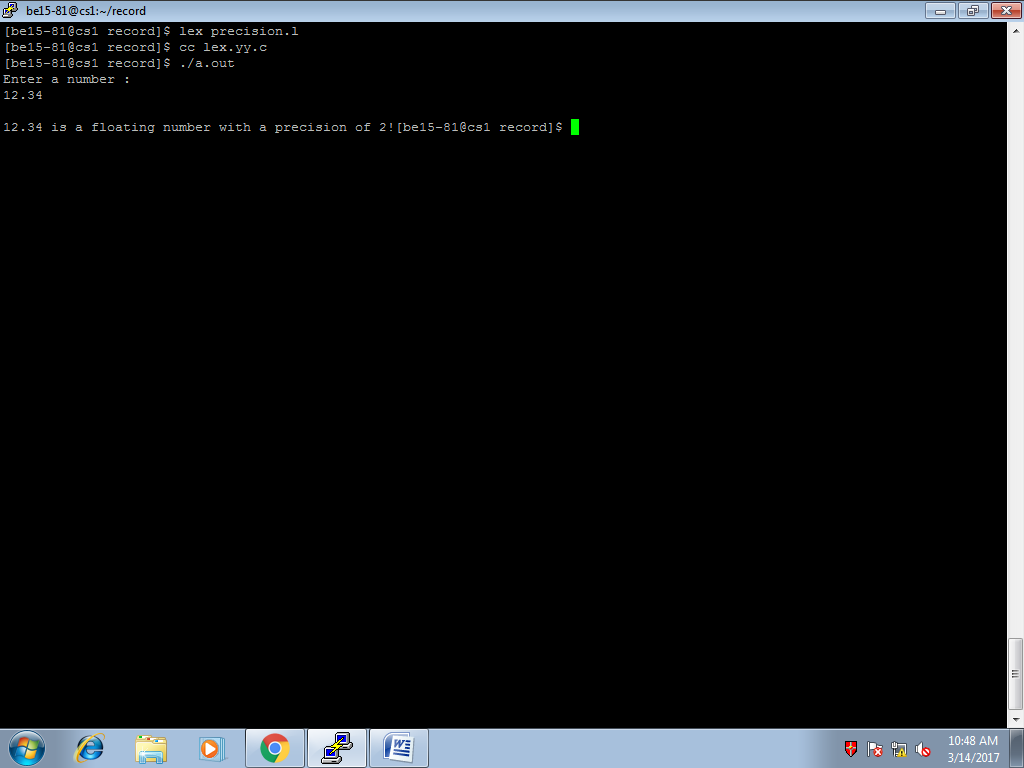
}

int yywrap()

{

return 1;

}



%{

#include<stdio.h>

int vowel=0;

int cons=0;

%}

%%

"a"|"e"|"i"|"o"|"u"|"A"|"E"|"I"|"O"|"U" {printf("%s vowel\n",yytext);vowel++;}

[a-zA-Z] {printf("%s consonant\n",yytext);cons++;}

[\n] { return 0;}

%%

int yywrap()

{

return 1;

}

int main()

{

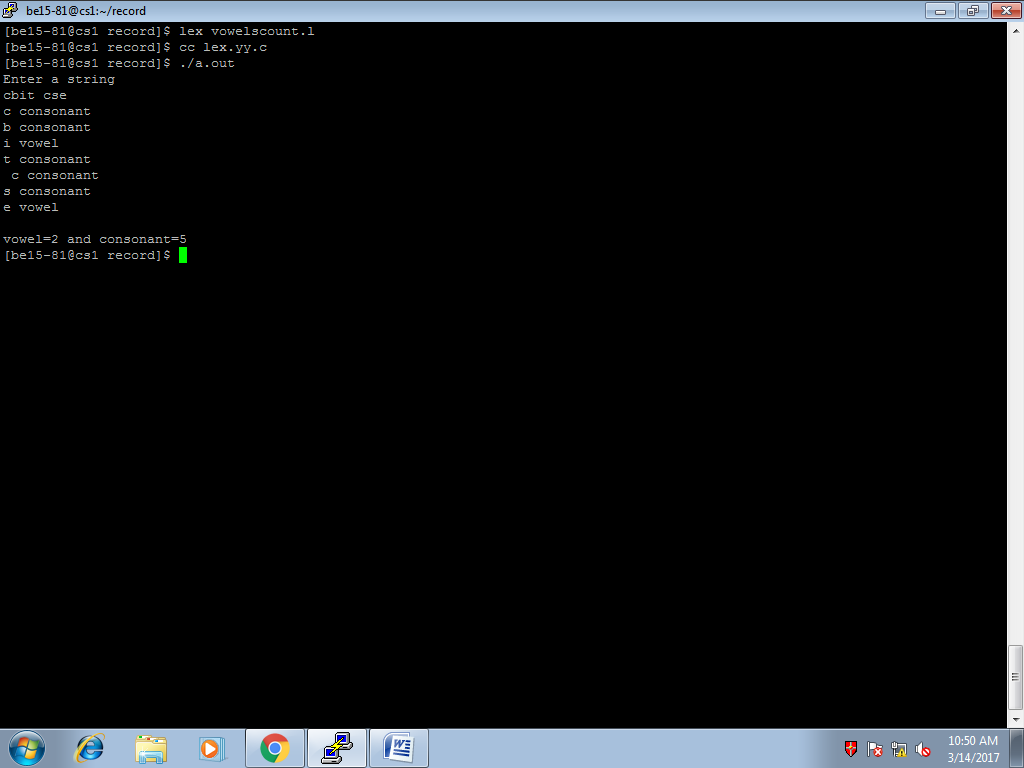
printf("Enter a string\n");

yylex();

printf("\nvowel=%d and consonant=%d\n",vowel,cons);

return 0;

}



//calculator

Calc.l

%{

#include<stdio.h>

#include "y.tab.h"

%}

%%

[0-9]+ {yylval.dval = atoi(yytext); return DIGIT;}

\n|. return yytext[0];

%%

Calc.y

%{

#include<stdio.h>

/\*E->E+E|E\*E|(E)|DIGIT\*/

%}

%union

{

int dval;

}

%token <dval> DIGIT

%type <dval> expr

%type <dval> expr1

%%

line : expr '\n' {printf("%d\n",$1);}

;

expr : expr '+' expr1 {$$ = $1 + $3 ;}

| expr '-' expr1 {$$ = $1 - $3 ;}

| expr '\*' expr1 {$$ = $1 \* $3 ;}

| expr '/' expr1 {$$ = $1 / $3 ;}

| expr1

;

expr1 : '('expr')' {$$=$2;}

| DIGIT

;

%%

int main()

{

yyparse ();

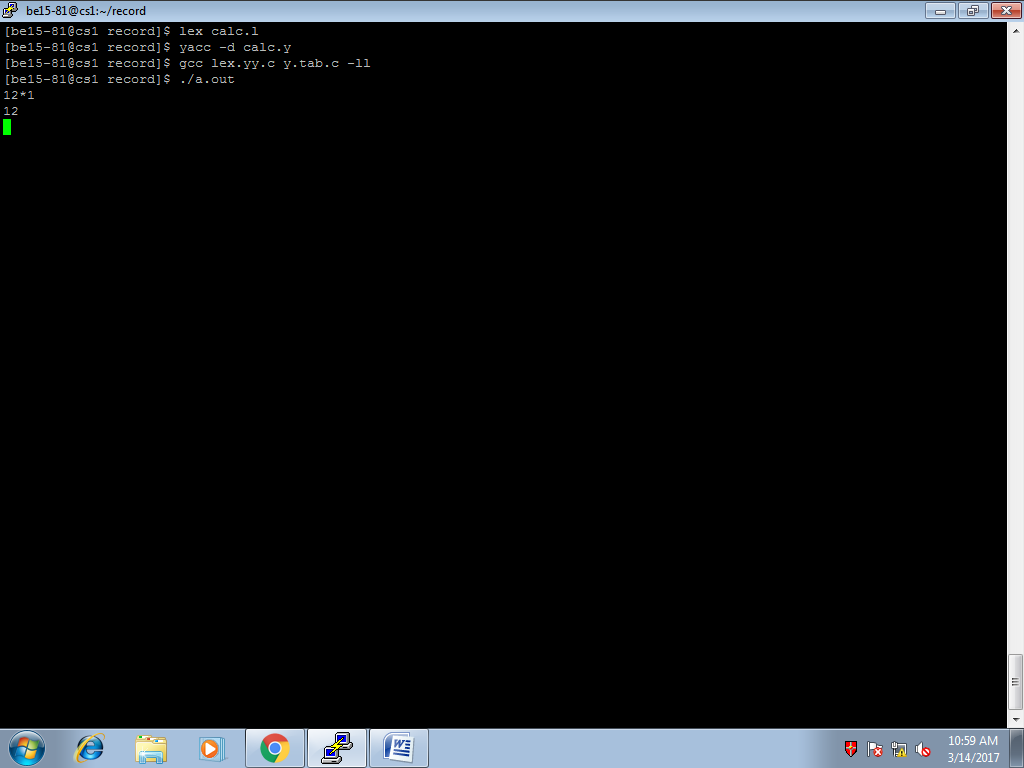
}

yyerror(char \*s)

{

printf("%s",s);

}



rstring.l

%{

#include<stdio.h>

#include"y.tab.h"

%}

%%

a return A;

b return B;

\n|. return yytext[0];

%%

Rstring.y

%{

#include<stdio.h>

int vd;

%}

%union

{

char dval;

}

%token <dval> A

%token <dval> B

%%

str: s '\n' { vd=1; return 0;}

s : A s B ;

| ;

%%

int main()

{

printf("enter the string");

yyparse();

if(vd==1)

printf(" valid");

else

printf(" not valid");

}

yyerror(char \*s)

{

printf("%s",s);

}

