**PROGRAM: Lex code to determine whether input is an identifier or not.**

**INPUT:**

%option noyywrap

%{

#include<stdio.h>

%}

%%

^[a-zA-Z\_][a-zA-Z0-9\_]\* printf("Valid Identifier.");

^[^a-zA-Z\_][a-zA-Z0-9\_]\* printf("Invalid Identifier.");

.;

%%

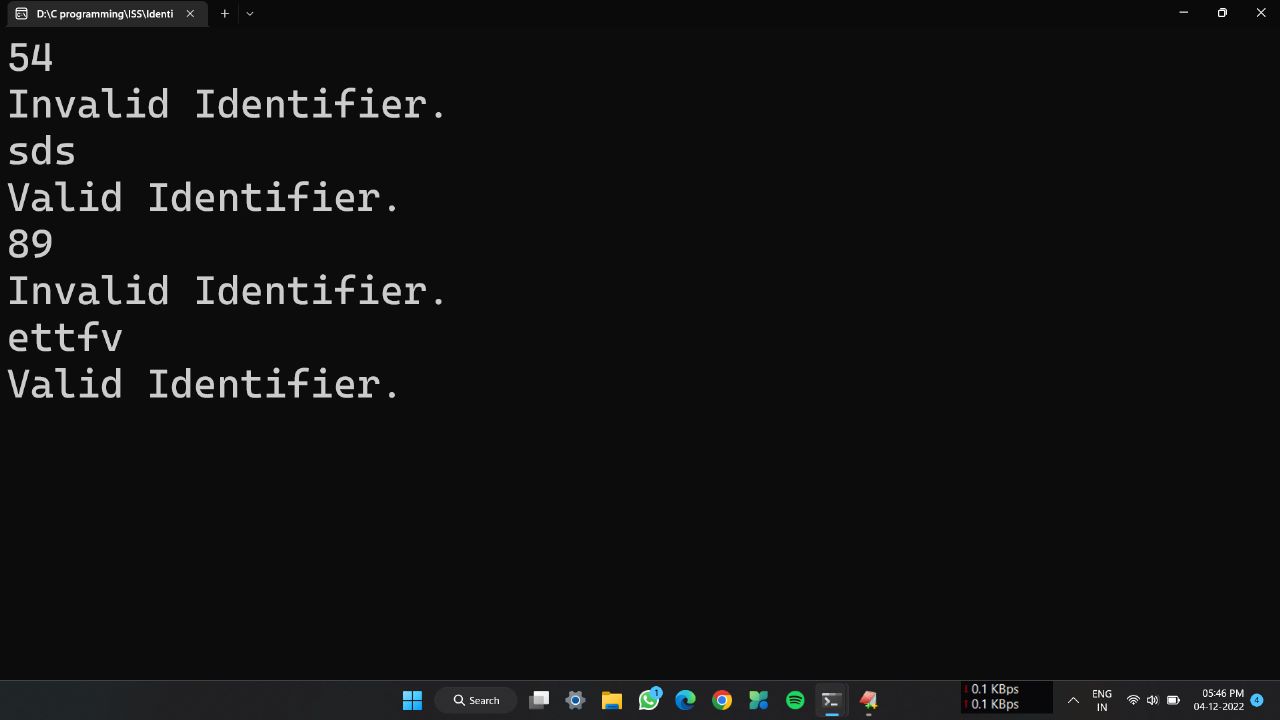
main()

{

yylex();

}

**OUTPUT:**



**PROGRAM: Lex program to check whether input is digit or not.**

**INPUT:**

%option noyywrap

%{

#include<stdio.h>

#include<stdlib.h>

%}

%%

[0-9]\* printf("Digit.");

^[^0-9] printf("Not a digit.");

.;

%%

int main()

{

yylex();

return 0;

}

**OUTPUT:**

****

**PROGRAM: Lex program to check whether the given number is even or odd.**

**INPUT:**

%{

#include<stdio.h>

int i;

%}

%%

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

if(i%2==0)

printf("Even");

else

printf("Odd");

}

%%

int yywrap(){}

int main()

{

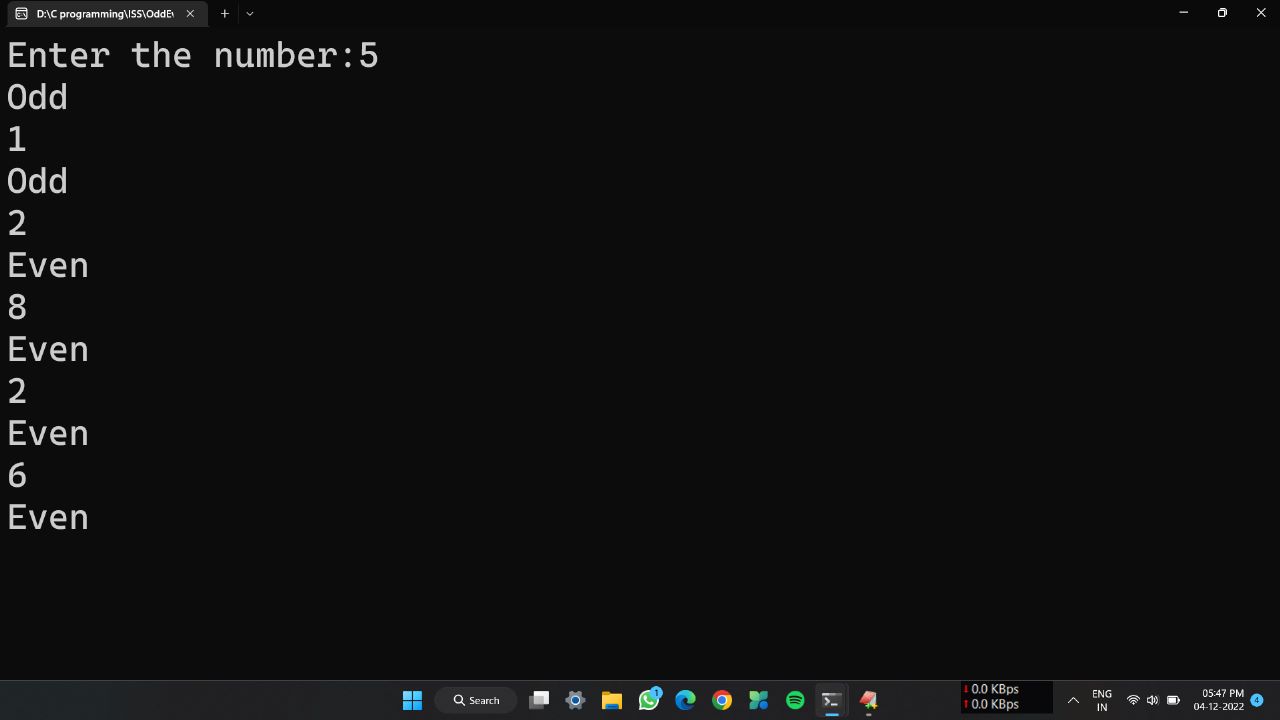
printf("Enter the number:");

yylex();

return 0;

}

**OUTPUT:**

****

**PROGRAM: Lex program to check whether a number is prime or not.**

**INPUT:**

%{

#include<stdio.h>

#include<stdlib.h>

int flag,c,j;

%}

%%

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

if(c==2)

{

printf("\n Prime Number.");

}

else if(c==0||c==1)

{

printf("\n Not a prime number.");

}

else

{

for(j=2;j<c;j++)

{

if(c%j==0);

flag=1;

}

if(flag==1)

printf("\n Not a prime number.");

else if(flag==0)

printf("\n Prime number.");

}

}

%%

int yywrap(){}

int main()

{

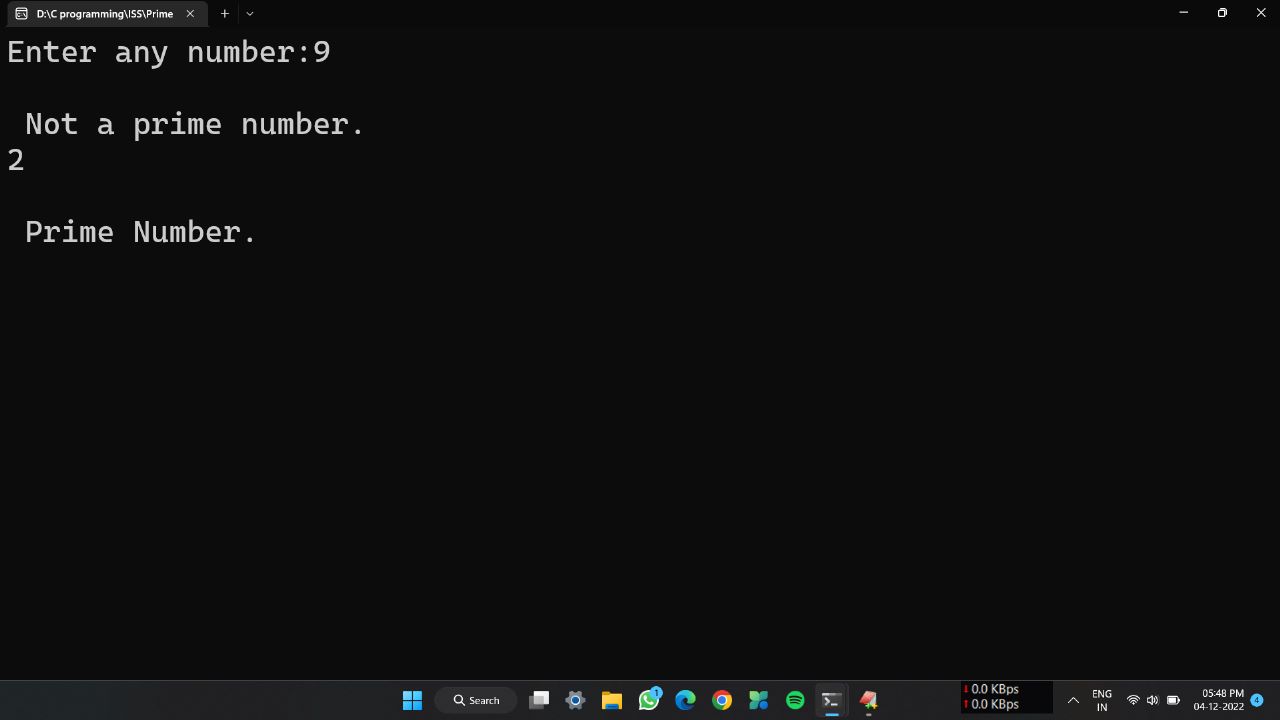
printf("Enter any number:");

yylex();

return 0;

}

**OUTPUT:**

****

**PROGRAM: Lex program to count total number of tokens.**

**INPUT:**

%{

int n=0;

%}

%%

"while"|"if"|"else" {n++;printf("\t keywords :%s",yytext);}

"int"|"float" {n++;printf("\tkeywords: %s",yytext);}

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

"<="|"=="|"="|"++"|"-"|"\*"|"+" {n++;printf("\t operator: %s",yytext);}

[(){}|,;] {n++;printf("\t separator: %s",yytext);}

[0-9]\*"."[0-9]+ {n++;printf("\t float: %s",yytext);}

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

.;

%%

int yywrap(){}

int main()

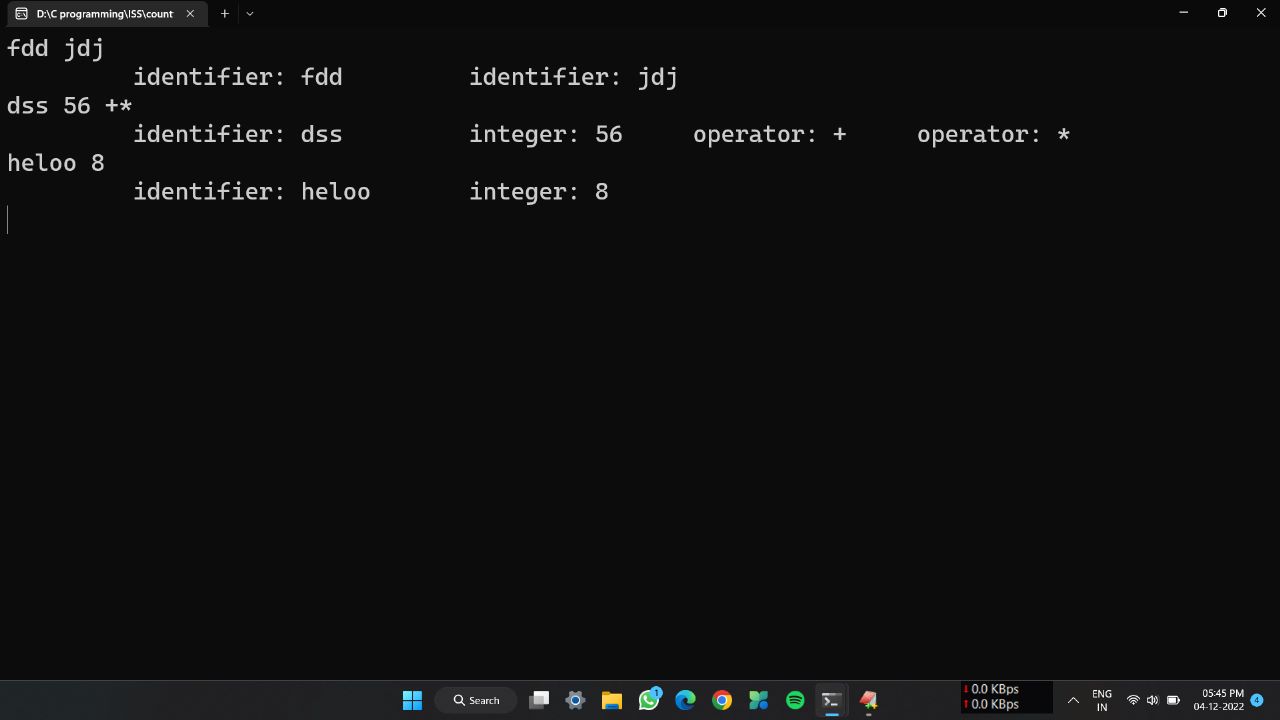
{

yylex();

printf("\n total no. of token=%d\n",n);

}

**OUTPUT:**

****

**PROGRAM: Lex program to implement a single calculation.**

**INPUT:**

%{

#include<stdio.h>

int op=0,i;

float a,b;

%}

dig [0-9]+|([0-9]\*)"."([0-9]+)

add "+"

sub "-"

mul "\*"

div "/"

pow "^"

ln \n

%%

{dig} {digi();}

{add} {op=1;}

{sub} {op=2;}

{mul} {op=3;}

{div} {op=4;}

{pow} {op=5;}

{ln} {printf("\n The answer:%f\n\n",a);}

%%

digi()

{

if(op==0)

a=atof(yytext);

else

{

b=atof(yytext);

switch(op)

{

case 1:a=a+b;

break;

case 2:a=a-b;

break;

case 3:a=a\*b;

break;

case 4:a=a/b;

break;

case 5:for(i=a;b>1;b--)

a=a\*i;

break;

}

op=0;

}

}

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

{

yylex();

}

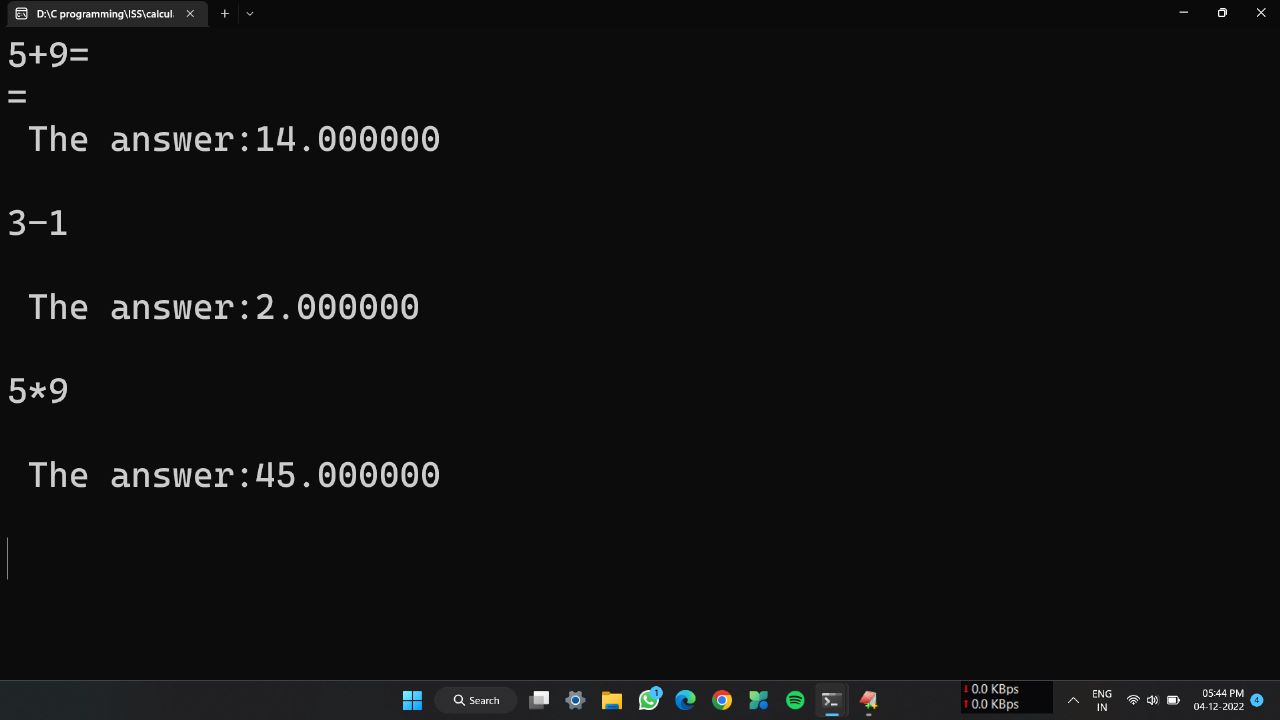
yywrap()

{

return 1;

}

**OUTPUT:**

****

**PROGRAM: Lex program to count no of words.**

**INPUT:**

%{

#include<stdio.h>

#include<string.h>

int i=0;

%}

%%

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

"\n" {printf("%d\n",i); i=0;}

%%

int yywrap(void){}

int main()

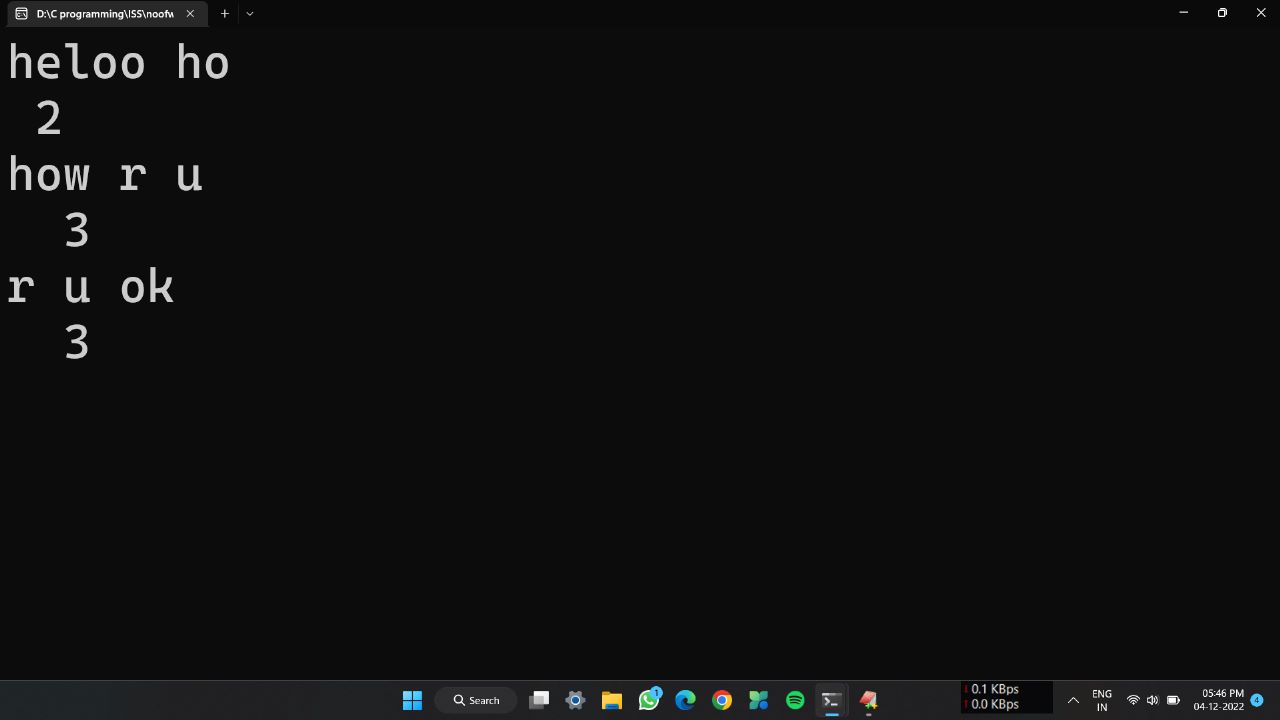
{

yylex();

return 0;

}

**OUTPUT:**

****