ARCHIVO DE DEFINICIÓN FLEX

```
%{
#include<stdio.h>
#include "y.tab.h" /* GENERADO AUTOMÁTICAMENTE POR BISON */
int bandera=0;
%}
%option noyywrap
%x recoparit recopinc recopdec
separador ([ \t""])+

letra [a-zA-Z]

digito [0-9]
digitos {digito}+
signo "+"|"-"
identificador {letra}({letra}|{digito})*("_")?({letra}|{digito})*
numero {signo}?{digitos}("."{digitos})?(E{signo}?{digitos})?
opasig "="|"+="|"-="|"*="|"/="|"%="
oparit "*"|"/"|"+"|"-"
oprel "<"|">"|"<="|">"|"!="|"=="
opinc "++"
opdec "--"
palres "#include"|"#define"|"for"|"return"|"switch"
palres1 "if"|"else"
palres2 "do"|"while"
palres3 "case"|"break"|"switch"
tipo  "void"|"int"|"float"|"char"|"long"|"double"
%%

      if(bandera==1) BEGIN(recoparit);
      else
      if(bandera==2) BEGIN(recopinc);
      else
      if(bandera==3) BEGIN(recopdec);
      else
      if(bandera==4) {bandera=0; BEGIN(INITIAL);}
{separador} {/* omitir */}
{identificador}/({opinc}";") {bandera=2; return(ID);}
{identificador}/({opdec}";") {bandera=3; return(ID);}
{identificador}/{oparit} {bandera=1; printf("Identificador: %s
\n",yytext); return(ID);}
{numero}/{oparit} {bandera=1; printf("Numero: %s\n",yytext);
return(NUM);}
")"/{oparit} {bandera=1; printf("Delimitador: %s\n",yytext);
return(DELIMPC);}
<recoparit>{oparit} {bandera=4; printf("Operador: %s\n",yytext);
return(OPARIT);}
<recopinc>{opinc} {bandera=4; printf("Incremento: %s\n",yytext);
return(OPASIGSUM);}
<recopdec>{opdec} {bandera=4; printf("Decremento: %s\n",yytext);
return(OPASIGRES);}

{numero} {return (NUM);}
```

```
"," {return (COMA);}
{opasig} {return (OPASIG);}
"++" {return (OPASIGSUM);}
"--" {return (OPASIGRES);}
{oparit} {return (OPARIT);}
"<"/{identificador}".h" {return (DELIMHI);}
">"$ {return (DELIMHD);}
{tipo}     {rerturn (TIPO);}
{oprel} {return (OPREL);}
{palres} {return (PALRES);}
{palres1} {return (PALRES1);}
{palres2} {return (PALRES2);}
{identificador} {return (ID);}
"(" {return (DELIMPA);}
")" {return (DELIMPC);}
";" {return (PC);}
"." {return (PUNTO);}
"{" {return (DELIMLA);}
"}" {return (DELIMLC);}
\n {return (SL);}
. ECHO;
%%
```

ARCHIVO DE DESCRIPCIÓN DE GRAMÁTICA DE BISON

```
%{
#include <stdio.h>
extern int yylex(void);
extern char *yytext;
extern FILE *yyin;
void yyerror(char *s);
%}
%token PALRES1 PALRES2 CASE BREAK RETURN SWITCH COMA ID NUM FRACC
NOT OPASIG OPASIGSUM OPASIGRES OPARIT DELIMHI DELIMLA DELIMLC
DELIMHD DELIMPA TIPO DELIMPC PUNTO SL OPREL PALRES PC
%%

instrucciones:  instrucciones instruccion
//    {printf("Se encontro una cadena de lenguaje valida\n");}
           |instruccion
//{printf("Se encontro una cadena de lenguaje valida\n");}
           ;
instruccion:    ID OPASIG expresion PC SL {printf("Se encontro
una instruccion de asignacion\n");}
           |ID OPASIGSUM PC SL {printf("Se encontro una
instruccion de incremento\n");}
           |ID OPASIGRES PC SL {printf("Se encontro una
instruccion de decremento\n");}
           |expresion PC SL
           |expresion SL
           |rutina SL
           |condicional SL
           |dowhile SL
           |while SL
           |for SL
           |switch SL
           ;
rutina:         listaparams DELIMPA listaparams DELIMPC
{printf("Inicio de rutina\n");} DELIMLA SL instrucciones PALRES
termino PC SL DELIMLC {printf("Fin de rutina\n");}
           ;
listaparams:    listaparams COMA param
           |param
           |listaparams COMA ID
           ;
param:          TIPO expresion
           ;
expresion:      termino
           |expresion OPASIG termino {printf("Se encontro una
instruccion de asignacion\n");}
           |expresion OPREL termino {printf("Se encontro una
instruccion de operacion relacional\n");}
           |expresion OPARIT termino {printf("Se encontro una
operacion aritmetica\n");}
           |encabezado
           |listaparams
           ;
```

```
termino: ID
        |NUM
      |FRACC
      |NOT
        |DELIMPA expresion DELIMPC
      |DELIMHI expresion DELIMHD
      ;
encabezado:     PALRES DELIMHI ID PUNTO ID DELIMHD {printf("Se
encontro un encabezado\n");}
           |PALRES ID PUNTO ID {printf("Se encontro un encabezado
\n");}
           |PALRES ID termino SL {printf("Se encontro un
encabezado\n");}
           ;
condicional:    PALRES1 expresion DELIMLA SL instrucciones
DELIMLC SL PALRES1 DELIMLA SL instrucciones DELIMLC {printf("Se
encontro una instruccion de un condicional if\n");}
           ;
dowhile:   PALRES2 DELIMLA SL instrucciones DELIMLC SL PALRES2
expresion PC {printf("Se encontro una instruccion de un ciclo do
while\n");}
           ;
while:          PALRES2 expresion DELIMLA SL instrucciones
DELIMLC SL {printf("Se encontro una instruccion de un ciclo while
\n");}
           ;
for:       PALRES DELIMPA expresion PC expresion PC expresion
DELIMPC DELIMLA SL instrucciones DELIMLC {printf("Se encontro una
instruccion de un ciclo for\n");}
           ;
switch:         PALRES expresion DELIMLA SL PALRES termino SL
instrucciones DELIMLC {printf("Se encontro una instruccion de
condicional switch\n");}
           ;
%%
void yyerror(char *s)
{
 printf("%s\n",s);
}

int main(int argc, char **argv)
{
 if(argc>1)
   yyin=fopen(argv[1],"rt");
 else
   yyin=fopen("entrada.txt","rt");
 yyparse();
 return 0;
}
```

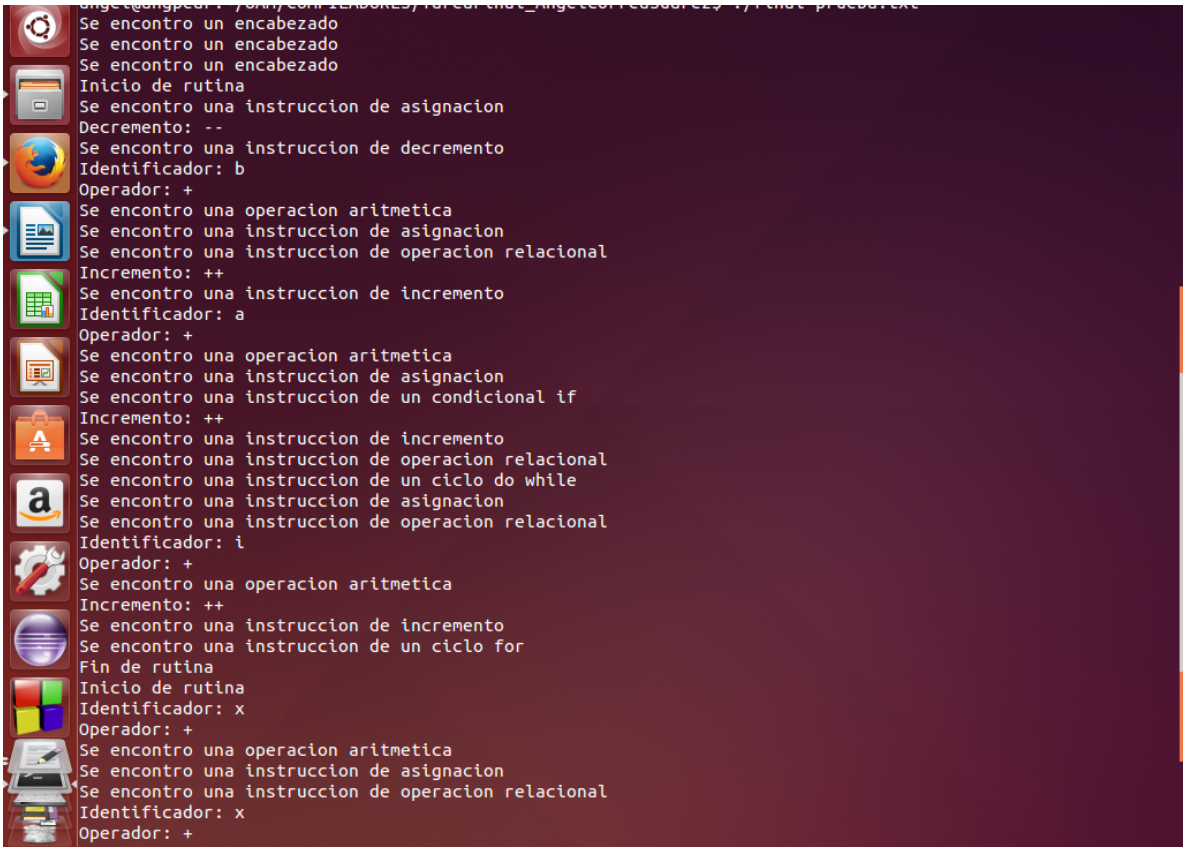
ARCHIVO DE PRUEBA DEL ANALIZADOR LÉXICO/SINTÁCTICO

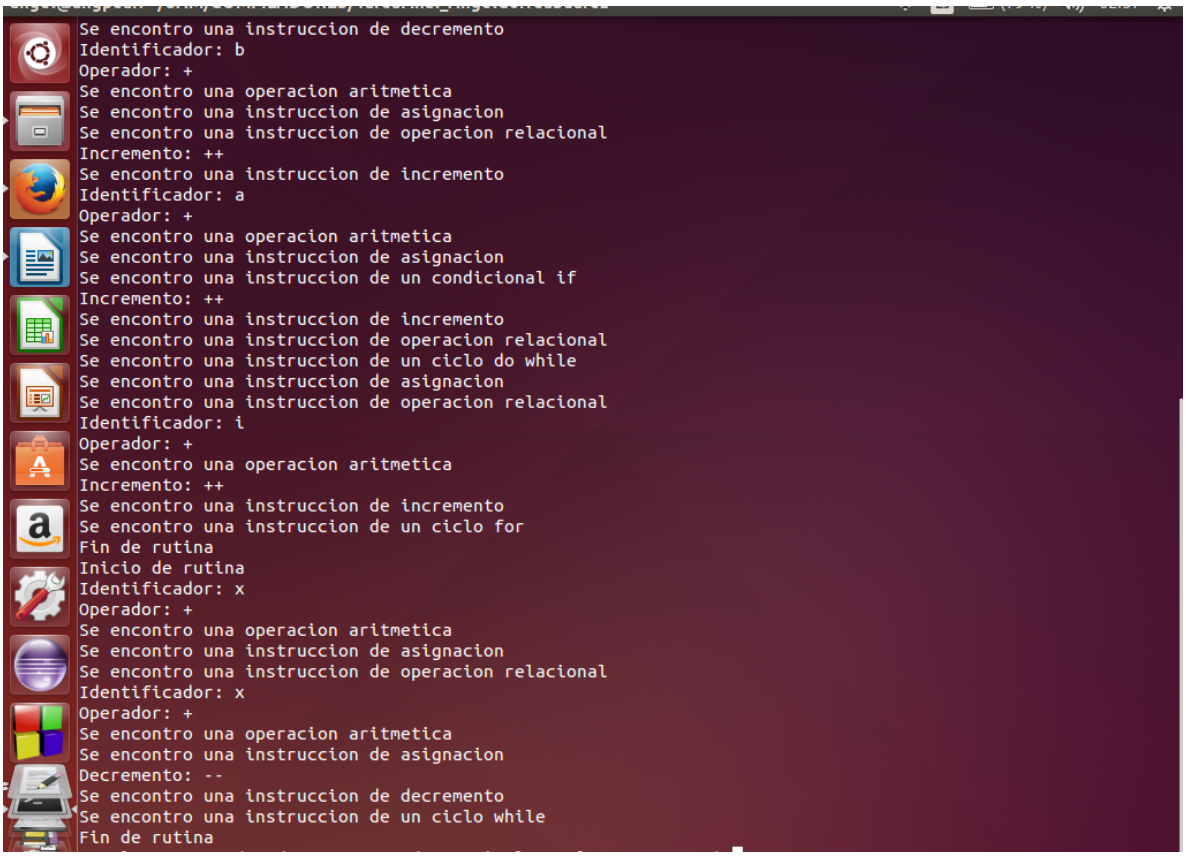
```
#include<stdio.h>
#include math.h
#define MAX 191

void main(int x){
      int i;
      float x,y,z;
      x=5;
      z--;
      a=b+c;
      if(x==10){
            x++;
      }
      else{
            a=a+1;
      }
      do{
            b++;
      }
      while(b<20);
      for(i=1; i<z; i+1){
            y++;
      }
return 1;
}
int funcion(int a, int b){
      double x,y,z;
      int i;
      i=x+5;
      while(z>y){
            z=x+y;
            x--;
      }

return 1;
}
```

PANTALLAS DE EJECUCIÓN DEL ANALIZADOR LÉXICO/SINTÁCTICO


Se encontro un encabezado
Se encontro un encabezado
Se encontro un encabezado
Inicio de rutina
Se encontro una instruccion de asignacion
Decremento: --
Se encontro una instruccion de decremento
Identificador: b
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Se encontro una instruccion de operacion relacional
Incremento: ++
Se encontro una instruccion de incremento
Identificador: a
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Se encontro una instruccion de un condicional if
Incremento: ++
Se encontro una instruccion de incremento
Se encontro una instruccion de operacion relacional
Se encontro una instruccion de un ciclo do while
Se encontro una instruccion de asignacion
Se encontro una instruccion de operacion relacional
Identificador: i
Operador: +
Se encontro una operacion aritmetica
Incremento: ++
Se encontro una instruccion de incremento
Se encontro una instruccion de un ciclo for
Fin de rutina
Inicio de rutina
Identificador: x
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Se encontro una instruccion de operacion relacional
Identificador: x
Operador: +


Se encontro una instruccion de decremento
Identificador: b
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Se encontro una instruccion de operacion relacional
Incremento: ++
Se encontro una instruccion de incremento
Identificador: a
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Se encontro una instruccion de un condicional if
Incremento: ++
Se encontro una instruccion de incremento
Se encontro una instruccion de operacion relacional
Se encontro una instruccion de un ciclo do while
Se encontro una instruccion de asignacion
Se encontro una instruccion de operacion relacional
Identificador: i
Operador: +
Se encontro una operacion aritmetica
Incremento: ++
Se encontro una instruccion de incremento
Se encontro una instruccion de un ciclo for
Fin de rutina
Inicio de rutina
Identificador: x
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Se encontro una instruccion de operacion relacional
Identificador: x
Operador: +
Se encontro una operacion aritmetica
Se encontro una instruccion de asignacion
Decremento: --
Se encontro una instruccion de decremento
Se encontro una instruccion de un ciclo while
Fin de rutina