# “COMPILADOR”

## INGENIERÍA EN SISTEMAS COMPUTACIONALES

PRESENTAN:

**ALEJANDRA LÓPEZ VEGA**

**RICARDO MORALES PARTIDA**

**CARLOS DANIEL OREGEL**

**DAVID JUNIOR RODRÍGUEZ FARIAS**

JIQUILPAN, MICHOACÁN, DICIEMBRE DE 2022

**Descripción general:**

Nuestra gramática se desarrolla de la siguiente forma:

En la parte de encima del programa se pueden declarar y asignar las variables tantas como así lo desee el usuario, y bien este puede solo declarar las variables o también asignarles valores.

Ya que después de este se desarrollan las sentencias tanto aritméticas (Operaciones), condición (IF) o de bucle (FOR), y si el usuario así lo requiere este puede hacer IF's anidados o también se puede realizar con FOR's anidados. Y así hasta que el usuario quiera desarrollar su código.

**Gramática:**

G(L)={T,N,Z,S}

T={id, int, float, char, , , ;, +,-,/,*,(,),{,},<,>,=,!, num, numf, litchar, for, if, else }

N={P, Tipo, V, Sen, A, Exp, Term, F, T, E, I, Con, D, EL, For, in, C, ad}

Z={

P -> Tipo id V | Sen

Tipo -> int | float | char

V -> , id V | ; P

Sen -> A Sen | I Sen | For Sen | &

A -> id = Exp ;

Exp -> Term E

Term -> F T

E -> + Term E | - Term E | &

F -> id | ( Exp ) | num | numf | litchar

T -> * F T | / F T | &

I -> if ( con ) { Sen } EL

Con -> id D id

D -> < | > | <= | >= | == | !=

EL -> else { Sen } | &

For -> for ( in ; con ; ad ) { Sen }

in -> int C | C

C -> id = num

Ad -> id + + | id - - | + + id | - - id

}

S={Z}

**Producciones:**

| | |
|---|---|
| P | 0. P’ -> P |
| P | 1. P -> Tipo id V |
| P | 2. P -> Sen |
| P | 3. Tipo -> int |
| P | 4. Tipo -> float |
| P | 5. Tipo -> char |
| P | 6. V -> , id V |
| P | 7. V -> ; P |
| P | 8. Sen -> A Sen |
| P | 9. Sen -> **I** Sen |
| P | 10. Sen -> For Sen |
| P | 11. Sen -> & |
| P | 12. A -> id = Exp ; |
| P | 13. Exp -> Term E |
| P | 14. Term -> F T |
| P | 15. E -> + Term E |
| P | 16. E -> - Term E |
| P | 17. E -> & |
| P | 18. F -> id |
| P | 19. F -> ( Exp ) |
| P | 20. F -> num |
| P | 21. F -> numf |
| P | 22. F -> litchar |
| P | 23. T -> * F T |
| P | 24. T -> / F T |
| P | 25. T -> & |
| P | 26. I -> if ( con ) { Sen } EL |
| P | 27. Con -> id D id |
| P | 28. D -> < |
| P | 29. D -> > |
| P | 30. D -> <= |
| P | 31. D -> >= |
| P | 32. D -> == |
| P | 33. D -> != |

| | **Primeros** | **Siguientes** |
|---|---|---|
| **P’** | Int, float, char, id, if , for | $ |
| **P** | Int, float, char, id, if , for | $ |
| **Tipo** | Int, float, char | id |
| **V** | , , ; | $ |
| **Sen** | Id , if , for, & | $, } |
| **A** | Id | Id , if , for, $, } |
| **Exp** | Id, ( , num, numf, litchar | ; , ) |
| **Term** | Id, ( , num, numf, litchar | +,-, ; , ) |
| **E** | +,-, & | ; , ) |
| **F** | Id, ( , num, numf, litchar | *, / , +,-, ; , ) |
| **T** | *, / , & | +,-, ; , ) |
| **I** | If | Id , if , for, $, } |
| **Con** | id | ;,) |
| **D** | <, >,=,! | id |
| **EL** | else, & | Id , if , for, $, } |
| **For** | for | Id , if , for, $, } |
| **In** | Int,id | ; |
| **C** | id | ; |
| **ad** | id, +, - | ) |

| | |
|---|---|
| P | 34. EL -> else { Sen } |
| P | 35. EL -> & |
| P | 36. For -> for ( in ; con ; ad ) { Sen } |
| P | 37. in -> int C |
| P | 38. in -> C |
| P | 39. C -> id = num |
| P | 40. Ad -> id + + |
| P | 41. Ad -> id - - |
| P | 42. Ad -> + + id |
| P | 43. Ad -> - - id |

**Desarrollo de gramática LR:**

I0=Cerr(P’ → ●P)

={ P’ → ● P > I1

P → ● Tipo id V > I2

P → ● Sen > I3

Tipo → ● int > I4

Tipo → ● float > I5

Tipo → ●char > I6

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

I1=Cerr(P’ → P●)

=Ir-a (I0,P)

={ P’ → P ● > P0

}

I2 = Cerr(P → Tipo ● id V)

= Ir-a (I0, Tipo)

={ P → Tipo ● id V > I13

}

I3=Cerr(P → Sen ●)

=Ir-a (I0, Sen)

={ P → Sen ● > P2

}

I4 = Cerr(Tipo → int ● )

= Ir-a (I0, int)

={ Tipo → int ● > P3

}

I5=Cerr(Tipo → float●)

=Ir-a (I0,float)

={ Tipo → float● > P4

}

I6=Cerr(Tipo → char●)

=Ir-a (I0,char)

={ Tipo → char● > P5

}

**I7** = Cerr(Sen → A ● Sen)

= Ir-a (I0, A)

={ Sen → A ● Sen > I14

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

**I8** =Cerr(Sen → I ● Sen )

=Ir-a (I0,I)

={ Sen → I ● Sen > I15

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

**I9** = Cerr(Sen →For ● Sen )

= Ir-a (I0, For)

={ Sen →For ● Sen > I16

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12 }

**I10** = Cerr( A → id ● = Exp ; )

= Ir-a (I0, id)

={ A → id ● = Exp ; > I17

}

**I11** = Cerr( I → if ● ( con ) { Sen } EL)

= Ir-a (I0, if)

={ I → if ● ( con ) { Sen } EL > I18

}

**I12** = Cerr( For → for ● ( in ; con ; ad ) { Sen } )

= Ir-a (I0, for)

={ For → for ● ( in ; con ; ad ) { Sen } > I19

}

**I13** = Cerr( P → Tipo id ● V)

= Ir-a (I2, id)

={ P → Tipo id ● V > I20

V → ● , id V > I21

V → ● ; P > I22

}

**I14** = Cerr( Sen → A Sen ● )

= Ir-a (I7, Sen)

={ Sen → A Sen ● > P8

}

**I15** = Cerr( Sen → I Sen ● )

= Ir-a (I8, Sen)

={ Sen → I Sen ● > P9

}

**I16** = Cerr( Sen →For Sen ● )

= Ir-a (I9, Sen)

={ Sen →For Sen ● > P10

}

**I17** = Cerr( A → id = ● Exp ; )

= Ir-a (I10, = )

={ A → id = ● Exp ; > I23

Exp → ●Term E > I24

Term → ● F T > I25

F → ● id > I26

F → ● ( Exp ) > I27

F → ● num > I28

F → ● numf > I29

F → ● litchar > I30

}

**I18** = Cerr( I → if ( ● con ) { Sen } EL )

= Ir-a (I11, **(** )

={ I → if ( ● con ) { Sen } EL > I31

con → ● id D id > I32

}

**I19** = Cerr( For → for ( ● in ; con ; ad ) { Sen } )

= Ir-a (I12, **(** )

={ For → for ( ● in ; con ; ad ) { Sen } > I33

in→ ● int C > I34

in → ● C > I35

C → ● id = num > I36

}

**I20** = Cerr( P → Tipo id V ● )

= Ir-a (I13, V )

={ P → Tipo id V ● > P1

}

**I21** = Cerr( V → , ● id V)

= Ir-a (I13, **,** )

={ V → , ● id V > I37

}

**I22** = Cerr( V → ; ● P )

= Ir-a (I13, **;** )

={ V → ; ● P > I38

P → ● Tipo id V > I2

P → ● Sen > I3

Tipo → ● int > I4

Tipo → ● float > I5

Tipo → ●char > I6

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

**I23** = Cerr( A → id = Exp ● ; )

= Ir-a (I17, Exp )

={ A → id = Exp ● ; > I39

}

**I24** = Cerr( Exp →Term ● E )

= Ir-a (I17, Term )

={ Exp → Term ● E > I40

E → ● + Term E > I41

E → ● - Term E > I42

E → ● & > P17

}

**I25** = Cerr( Term → F ●T)

= Ir-a (I17, F)

={ Term → F● T > I43

T → ● * F T > I44

T → ● / F T > I45

T → ● & > P25

}

**I26** = Cerr( F → id ●)

= Ir-a (I17, id )

={ F → id ● > P18

}

**I27** = Cerr( F → ( ● Exp ) )

= Ir-a (I17, ( )

={ F → ( ● Exp ) > I46

Exp → ●Term E > I24

Term → ● F T > I25

F → ● id > I26

F → ● ( Exp ) > I27

F → ● num > I28

F → ● numf > I29

F → ● litchar > I30

}

**I28** = Cerr(F → num ●)

= Ir-a (I17, num )

={ F → num ● > P20

}

**I29** = Cerr( F → numf ● )

= Ir-a (I17, numf )

={ F → numf ● > P21

}

**I30** = Cerr(F → litchar ●)

= Ir-a (I17, litcar )

={ F → litchar ● > P22

}

**I31** = Cerr( I → if ( con ● ) { Sen } EL )

= Ir-a (I18, con )

={ I → if ( con ● ) { Sen } EL > I47 }

**I32** = Cerr(con → id ● D id )

= Ir-a (I18, id )

={ con → id ● D id >I48

D → ● < >I49

D → ● > >I50

D → ● <= >I51

D → ● >= >I52

D → ● == >I53

D → ● != >I54

}

**I33** = Cerr( For → for (in ● ; con ; ad ) { Sen } )

= Ir-a (I19, in)

={ For → for (in ● ; con ; ad ) { Sen } >I55

}

**I34** = Cerr( in→ int ● C )

= Ir-a (I19, int )

={ in→ int ● C > I56

C → ● id = num > I36 }

**I35** = Cerr( in → C ●)

= Ir-a (I19, C )

={ in → C ● > P38

}

**I36** = Cerr( C → id ● = num )

= Ir-a (I19, id )

={ C → id ● = num > I57

}

**I37** = Cerr( V → , id ● V)

= Ir-a (I21, id )

={ V → , id ● V > I58

V → ● , id V > I21

V → ● ; P > I22

}

**I38** = Cerr( V → ; P ●)

= Ir-a (I22, P )

={ V → ; P● > P7

}

**I39** = Cerr( A → id = Exp ; ● )

= Ir-a (I23, ;)

={ A → id = Exp ; ● > P12

}

**I40** = Cerr( Exp →Term E ●)

= Ir-a (I24, E)

={ Exp → Term E ● > P13

}

**I41** = Cerr( E → + ● Term E)

= Ir-a (I24, + )

={ E → + ● Term E > I59

Term → ● F T > I25

F → ● id > I26

F → ● ( Exp ) > I27

F → ● num > I28

F → ● numf > I29

F → ● litchar > I30

}

**I42** = Cerr( E → - ● Term E )

= Ir-a (I24, -)

={ E → - ● Term E > I60

Term → ● F T > I25

F → ● id > I26

F → ● ( Exp ) > I27

F → ● num > I28

F → ● numf > I29

F → ● litchar > I30

}

**I43** = Cerr( Term → F T●)

= Ir-a (I25, T)

={ Term → F T ● > P14

}

**I44** = Cerr( T → * ● F T )

= Ir-a (I25, * )

={ T → * ● F T > I61

F → ● id > I26

F → ● ( Exp ) > I27

F → ● num > I28

F → ● numf > I29

F → ● litchar > I30

}

**I45** = Cerr( T → / ● F T )

= Ir-a (I25, / )

={ T → / ● F T > I62

F → ● id > I26

F → ● ( Exp ) > I27

F → ● num > I28

F → ● numf > I29

F → ● litchar > I30

}

**I46** = Cerr( F → ( Exp ● ) )

= Ir-a (I27, Exp )

={ F → ( Exp ● ) > I63

}

**I47** = Cerr( I → if ( con ) ● { Sen } EL )

= Ir-a (I31, **)** )

={ I → if ( con ) ●{ Sen } EL > I64

}

**I48** = Cerr( con → id D ● id )

= Ir-a (I32, D )

={ con → id D ● id > I65

}

**I49** = Cerr( D → < ● )

= Ir-a (I32, < )

={ D → < ● > P28

}

**I50 =** Cerr(D → > ●)

= Ir-a (I32, > )

={ D → > ● > P29

}

**I51** = Cerr(D → < = ●)

= Ir-a (I32, <= )

={ D → <= ● > P30

}

**I52** = Cerr( D → >= ● )

= Ir-a (I32, >= )

={ D → > = ● > P31 }

**I53** = Cerr(D →  == ● )

= Ir-a (I32, == )

={      D →  == ●                                        > P32

}

**I54** = Cerr( D → != ●)

= Ir-a (I32, != )

={      D → != ●                                         > P33

}

**I55** = Cerr( For → for  (in  ; ● con ; ad ) { Sen } )

= Ir-a (I33, ;)

={      For → for (in ; ● con ; ad ) { Sen }          >I66

con →  ● id D id                                  >I32

}

**I56** = Cerr( in→ int C ● )

= Ir-a (I34, C )

={      in→ int C ●                                      > P37

}

**I57** = Cerr( C → id = ●  num )

= Ir-a (I36, = )

={      C → id = ●  num                                 > I67

}

**I58** = Cerr( V → , id  V ●)

= Ir-a (I37, V )

={      V → , id  V ●                                    > P6            }

**I59** = Cerr( E → + Term ● E )

= Ir-a (I41, Term )

={ E → + Term ● E > I68

E → ● + Term E > I41

E → ● - Term E > I42

E → ● & > P17

}

**I60** = Cerr( E → - Term ● E )

= Ir-a (I42, Term)

={ E → - Term ● E > I69

E → ● + Term E > I41

E → ● - Term E > I42

E → ● & > P17

}

**I61** = Cerr( T → * F ● T)

= Ir-a (I44, F )

={ T → * F ● T > I70

T → ● * F T > I44

T → ● / F T > I45

T → ● & > P25

}

**I62** = Cerr( T → / F ● T )

= Ir-a (I45, F )

={ T → / F ● T > I71

T → ● * F T > I44

T → ● / F T > I45

T → ● & > P25

}

**I63** = Cerr( F → ( Exp ) ● )

= Ir-a (I46, **)** )

={ F → ( Exp ) ● > P19

}

**I64 =** Cerr( I → if ( con ) {● Sen } EL)

= Ir-a (I47, **{** )

={ I → if ( con ) {● Sen } EL > I72

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

**I65** = Cerr( con → id D id ● )

= Ir-a (I48, id )

={ con → id D id ● > P27

}

**I66** = Cerr( For → for (in ; con ● ; ad ) { Sen }  )

= Ir-a (I55, con )

={        For → for (in ; con ● ; ad ) { Sen }         > I73

}

**I67** = Cerr( C → id =  num ● )

= Ir-a (I57, num )

={        C → id = num ●         > P39

}

**I68** = Cerr( E → +  Term  E ●)

= Ir-a (I60, E)

={       E → +  Term  E ●         > P15

}

**I69** = Cerr( E → -  Term  E ●)

= Ir-a (I60, E)

={        E → -  Term  E ●         > P16

}

**I70** = Cerr( T →  * F T● )

= Ir-a (I61, T )

={       T →  * F T●         > P23

}

**I71** = Cerr( T → /  F  T ● )

= Ir-a (I62, T )

={       T →  / F T ●         > P24

}

**I72** = Cerr( I → if ( con ) { Sen ● } EL )

= Ir-a (I64, Sen )

={ I → if ( con ) { Sen ● } EL > I74

}

**I73** = Cerr( For → for (in ; con ; ● ad ) { Sen } )

= Ir-a (I66, ; )

={ For → for (in ; con ; ● ad ) { Sen } > I75

ad → ● id ++ > I76

ad → ● id -- > I76

ad → ● ++ id > I77

ad → ● -- id > I78

}

**I74** = Cerr( I → if ( con ) { Sen }● EL)

= Ir-a (I72**, }** )

={ I → if ( con ) { Sen } ● EL > I79

EL → ● else { Sen } > I80

EL → ●& > P35

}

**I75** = Cerr( For → for (in ; con ; ad ●) { Sen } )

= Ir-a (I73, ad )

={ For → for (in ; con ; ad ● ) { Sen } > I81

}

**I76** = Cerr(ad → id ● ++

ad → id ● --) )

= Ir-a (I73, id )

={ ad → id ● ++ > I82

ad → id ● - - > I83

}

**I77** = Cerr( ad → ++ ● id)

= Ir-a (I73, ++)

={ ad → ++ ● id > I84

}

**I78** = Cerr( ad → -- ● id )

= Ir-a (I73, -- )

={ ad → -- ● id > I85

}

**I79** = Cerr( I → if ( con ) { Sen } EL ● )

= Ir-a (I74, EL )

={ I → if ( con ) { Sen } EL ● > P26

}

**I80** = Cerr( EL → else ● { Sen } )

= Ir-a (I74, else )

={ EL → else ● { Sen } > I86

}

**I81** = Cerr( For → for (in ; con ; ad ) ● { Sen } )

= Ir-a (I75, **)** )

={ For → for (in ; con ; ad ) ● { Sen } > I87

}

**I82** = Cerr(ad → id ++ ● )

= Ir-a (I76, ++ )

={ ad → id ++ ● > P40

}

**I83** = Cerr( ad → id - - ● )

= Ir-a (I76, - - )

={ ad → id - - ● > P41

}

**I84** = Cerr( ad → ++ id ●)

= Ir-a (I77, id)

={ ad → ++ id ● > P42

}

**I85** = Cerr( ad → - - id ●)

= Ir-a (I78, id )

={ ad → -- id ● > P43

}

**I86** = Cerr( EL → else { ● Sen } )

= Ir-a (I80, **{** )

={ EL → else { ● Sen } > I88

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

**I87** = Cerr( For → for (in ; con ; ad ) { ● Sen } )

= Ir-a (I81, { )

={ For → for (in ; con ; ad ) { ● Sen } > I89

Sen → ● A Sen > I7

Sen → ● I Sen > I8

Sen → ● For Sen > I9

Sen → ● & > P11

A → ● id = Exp ; > I10

I → ● if ( con ) { Sen } EL > I11

For → ● for ( in ; con ; ad ) { Sen } > I12

}

**I88** = Cerr( EL → else { Sen ● } )

= Ir-a (I86, **Sen** )

={ EL → else { Sen ● } > I90

}

**I89** = Cerr( For → for (in ; con ; ad ) { Sen ● } )

= Ir-a (I87, **Sen** )

={ For → for (in ; con ; ad ) { Sen ● } > I91

}

**I90** = Cerr( EL → else { Sen } ● )

= Ir-a (I88, **}** )

={ EL → else { Sen } ● > P34

}

**I91** = Cerr( For → for (in ; con ; ad ) { Sen } ●)

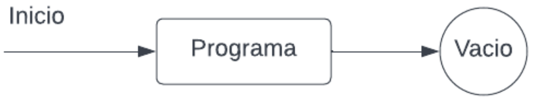
= Ir-a (I89, **}** )

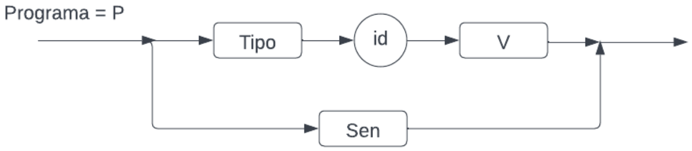
={ For → for (in ; con ; ad ) { Sen } ● > P36

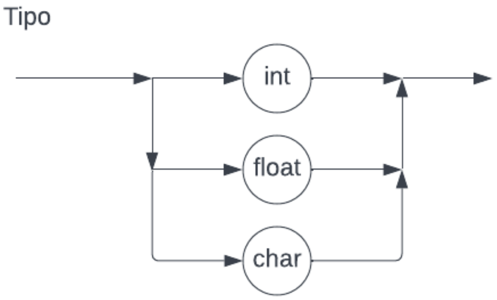
}

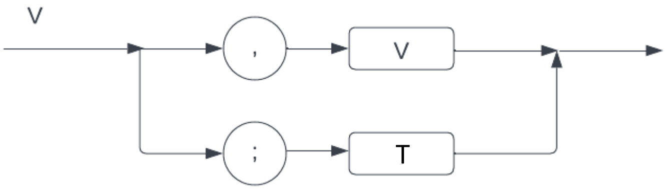
**Enlace de tabla de análisis sintáctica:**

## Grafo sintáctico

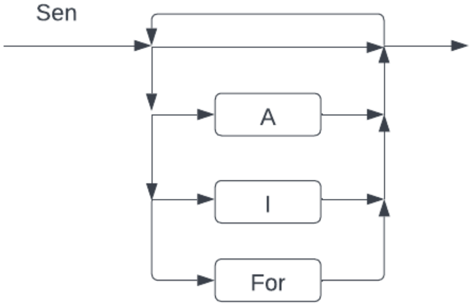




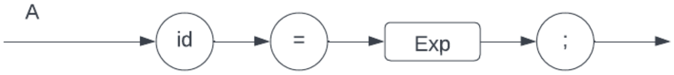




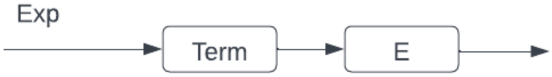
Sen



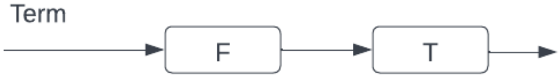
A



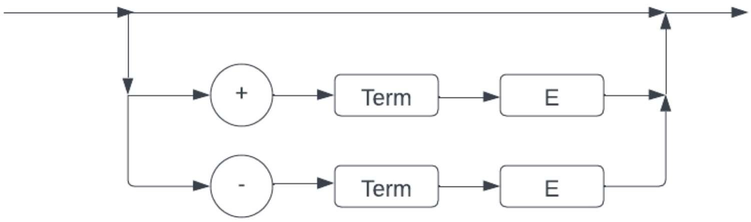
Exp



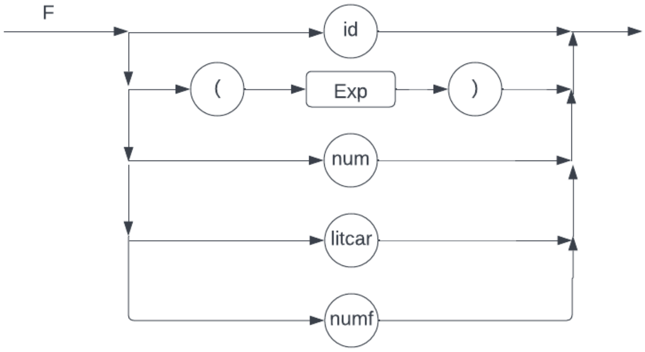
Term



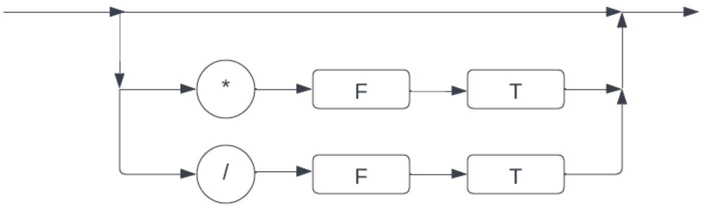
E

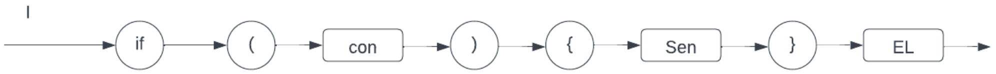


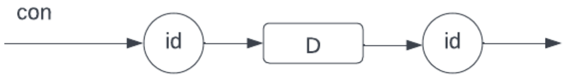
F

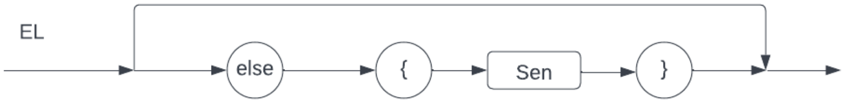


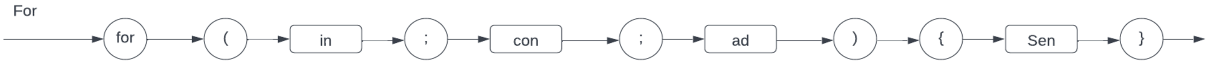
T

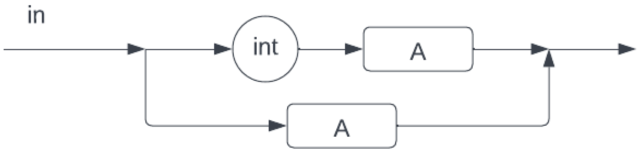


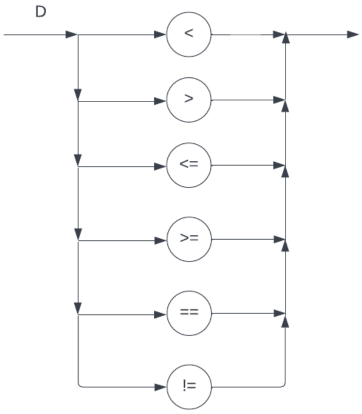

I
if
(
con
)
{
Sen
}
EL

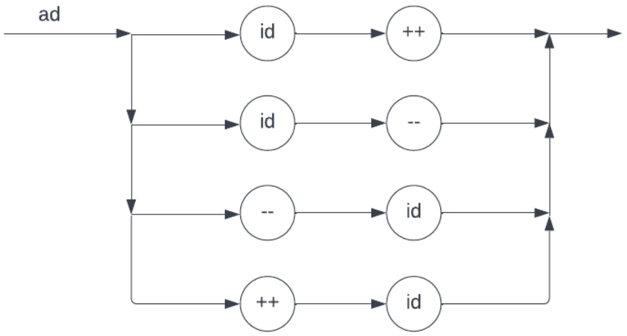

con
id
D
id


EL
else
{
Sen
}


For
for
(
in
;
con
;
ad
)
{
Sen
}


in
int
A
A


D
<
>
<=
>=
==
!=


ad
id
++
id
--
--
id
++
id

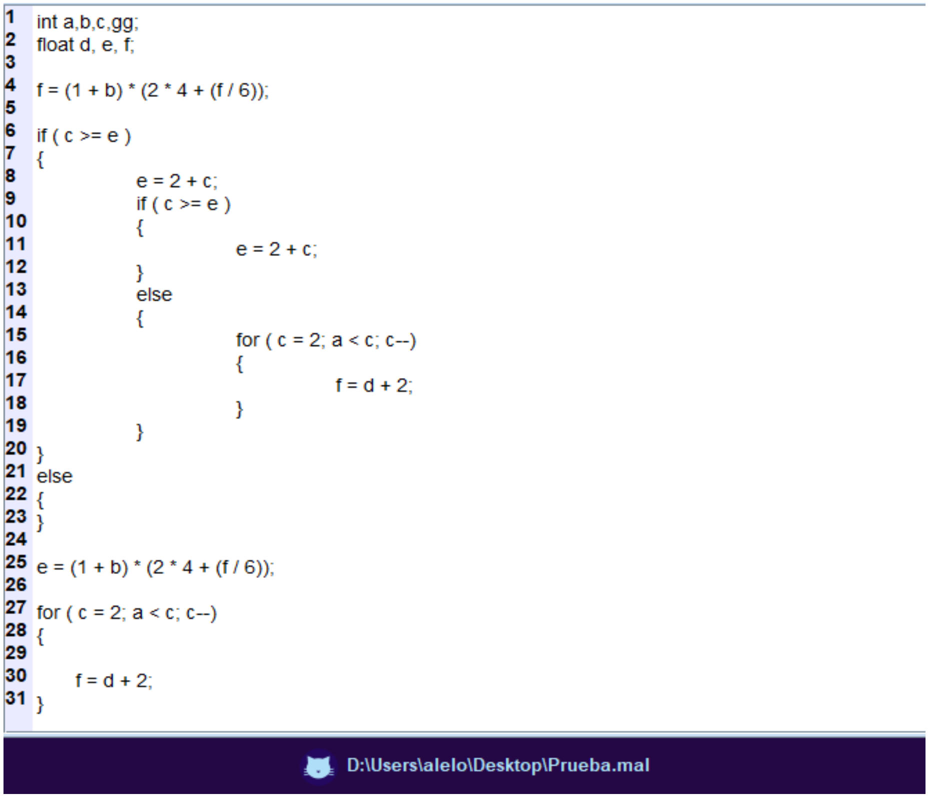
## Reglas semánticas

```
private int suma[][] = {
        //                  int 0    float 1   char2
        /*int*/          {0,       1,      -1},
        /*float*/        {1,       1,      -1},
        /*char*/         {-1,     -1,       2}};

private int rest[][] = {
        //                  int 0    float 1   char2
        /*int*/          {0,       1,       -1},
        /*float*/        {1,       1,       -1},
        /*char*/         {-1,     -1,       -1}};

private int divi[][] = {
        //                  int 0    float 1   char2
        /*int*/          {1,       1,       -1},
        /*float*/        {1,       1,       -1},
        /*char*/         {-1,     -1,       -1}};

private int mult[][] = {
        //                  int 0    float 1   char2
        /*int*/          {0,       1,       -1},
        /*float*/        {1,       1,       -1},
        /*char*/         {-1,     -1,       -1}};

private boolean igua[][] = {
        //                  int       float    char
        /*int*/          {true,      false,   false},
        /*float*/        {true,      true,    false},
        /*char*/         {false,     false,   true}};
```

```
private boolean condi[][] = {
        //              int 0   float 1   char2
        /*int*/         {true,   true,     false},
        /*float*/       {true,   true,     false},
        /*char*/        {false,  false,    true}};
```

## Capturas de pantalla

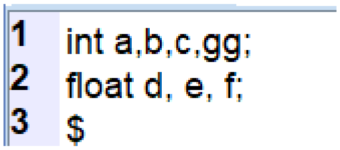
*Código*



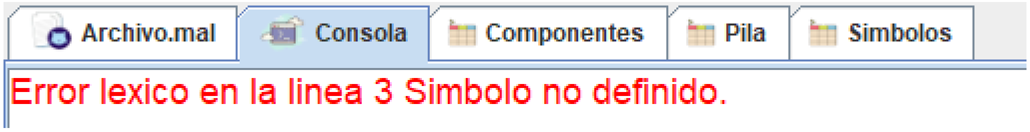
```
int a,b,c,gg;
float d, e, f;

f = (1 + b) * (2 * 4 + (f / 6));

if ( c >= e )
{
        e = 2 + c;
        if ( c >= e )
        {
                e = 2 + c;
        }
        else
        {
                for ( c = 2; a < c; c--)
                {
                        f = d + 2;
                }
        }
}
else
{
}

e = (1 + b) * (2 * 4 + (f / 6));

for ( c = 2; a < c; c--)
{

    f = d + 2;
}
```

*Errores*

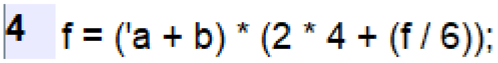
1. Léxicos



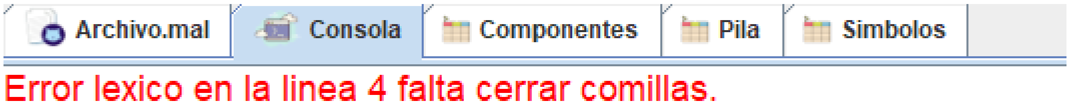
```
int a,b,c,gg;
float d, e, f;
$
```



```
Error lexico en la linea 3 Simbolo no definido.
```

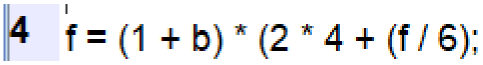


```
4   f = ('a + b) * (2 * 4 + (f / 6));
```

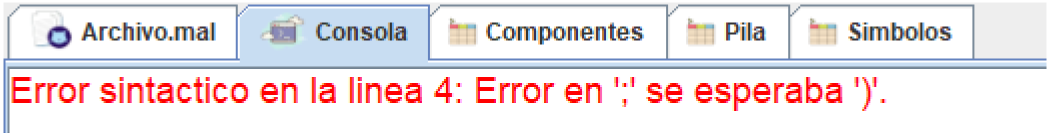


```
Error lexico en la linea 4 falta cerrar comillas.
```

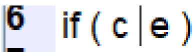
2. Sintácticos



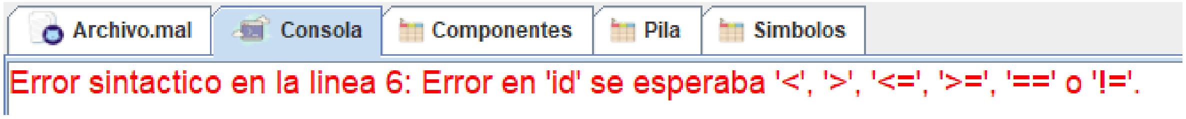
```
4   f = (1 + b) * (2 * 4 + (f / 6);
```



```
Error sintactico en la linea 4: Error en ';' se esperaba ')'.
```

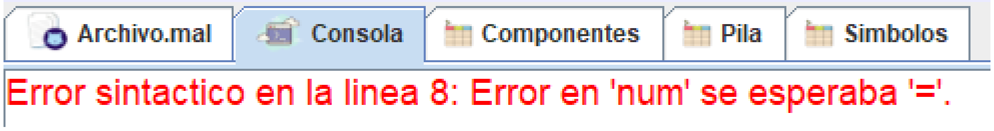


```
6   if ( c e )
```

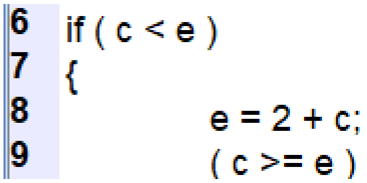


```
Error sintactico en la linea 6: Error en 'id' se esperaba '<', '>', '<=', '>=', '==' o '!='.
```

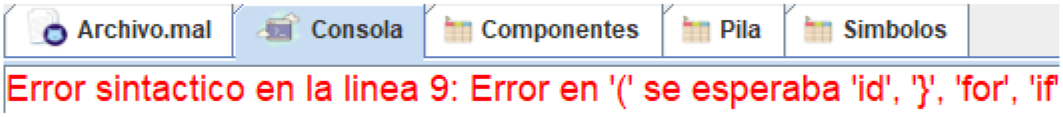
```
8               e  2 + c;
```



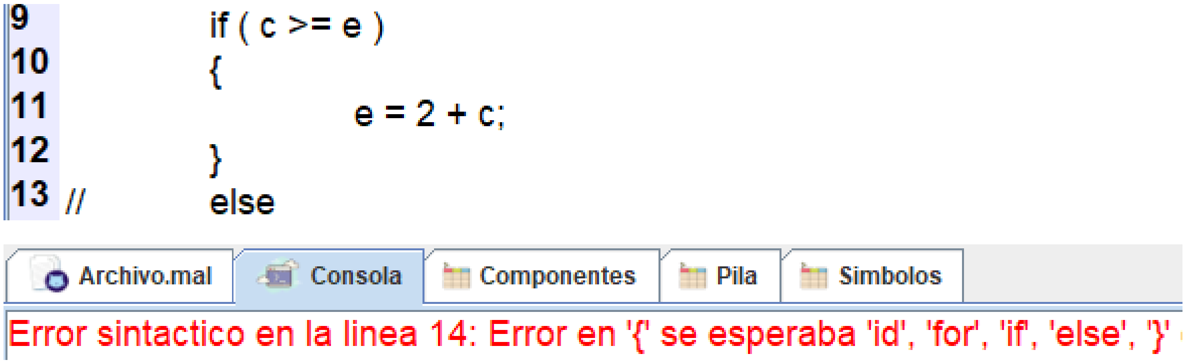
```
Error sintactico en la linea 8: Error en 'num' se esperaba '='.
```



```
6   if ( c < e )
7   {
8               e = 2 + c;
9               ( c >= e )
```

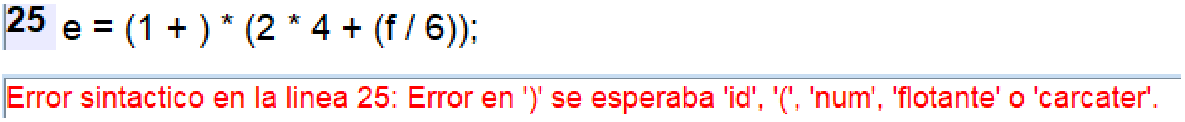


```
Error sintactico en la linea 9: Error en '(' se esperaba 'id', '}', 'for', 'if'
```



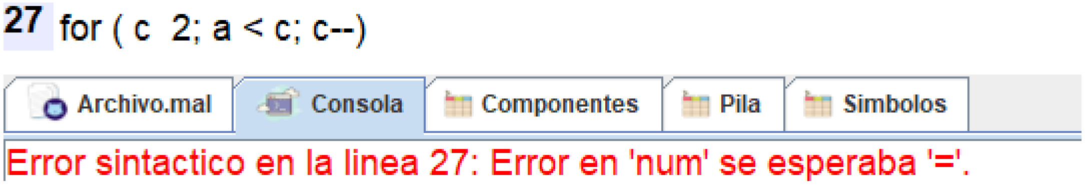
```
9            if ( c >= e )
10           {
11                   e = 2 + c;
12           }
13 //        else
```

```
Error sintactico en la linea 14: Error en '{' se esperaba 'id', 'for', 'if', 'else', '}'
```



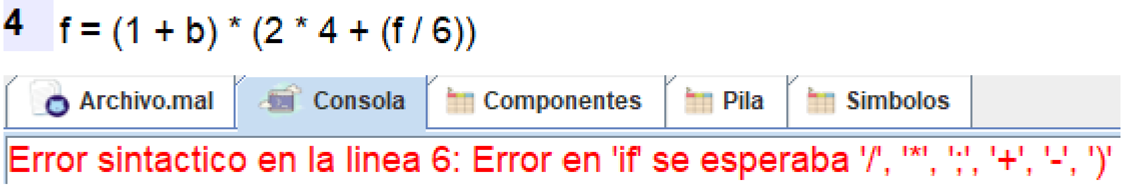
```
25 e = (1 + ) * (2 * 4 + (f / 6));
```

```
Error sintactico en la linea 25: Error en ')' se esperaba 'id', '(', 'num', 'flotante' o 'carcater'.
```



```
27 for ( c  2; a < c; c--)
```

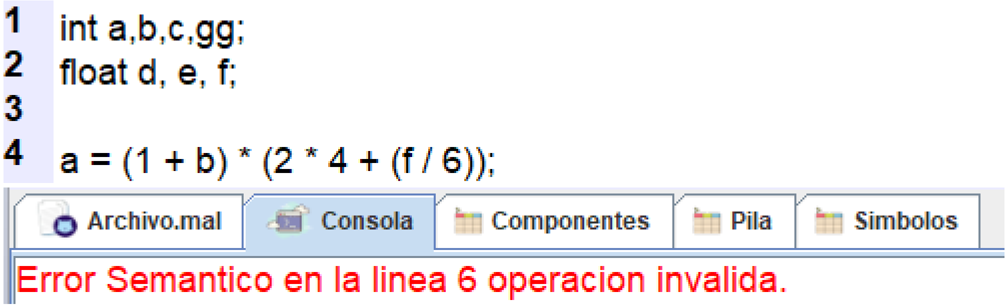
```
Error sintactico en la linea 27: Error en 'num' se esperaba '='.
```



```
4  f = (1 + b) * (2 * 4 + (f / 6))
```

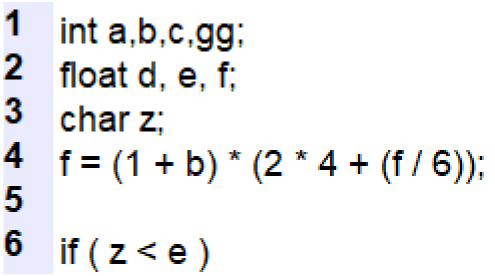
```
Error sintactico en la linea 6: Error en 'if' se esperaba '/', '*', ';', '+', '-', ')'
```

3. Semánticos

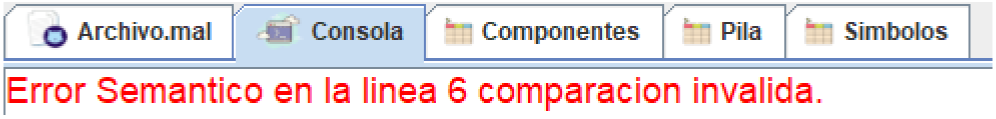


```
1  int a,b,c,gg;
2  float d, e, f;
3
4  a = (1 + b) * (2 * 4 + (f / 6));
```

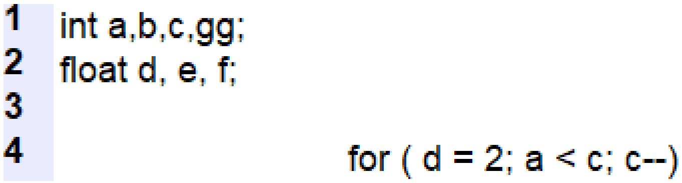
```
Error Semantico en la linea 6 operacion invalida.
```



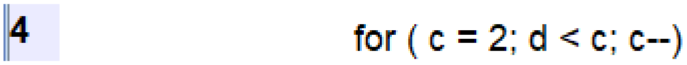
```
1  int a,b,c,gg;
2  float d, e, f;
3  char z;
4  f = (1 + b) * (2 * 4 + (f / 6));
5
6  if ( z < e )
```



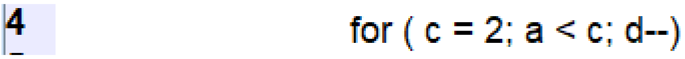
```
Error Semantico en la linea 6 comparacion invalida.
```



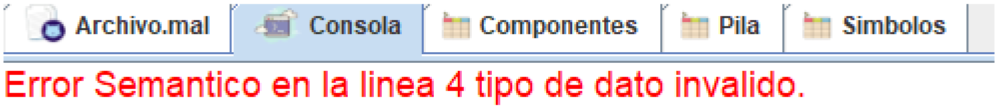
```
1  int a,b,c,gg;
2  float d, e, f;
3
4                       for ( d = 2; a < c; c--)
```



```
4                       for ( c = 2; d < c; c--)
```

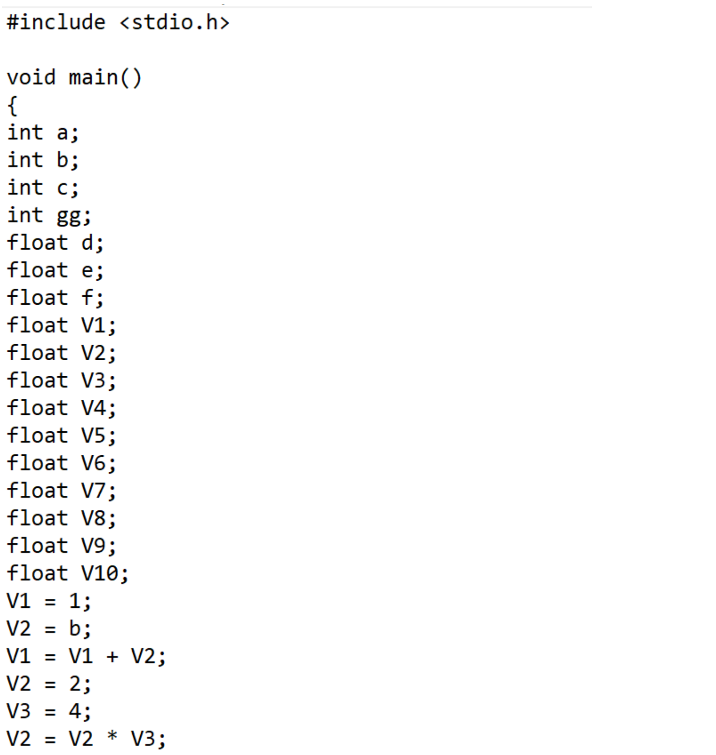


```
4                       for ( c = 2; a < c; d--)
```

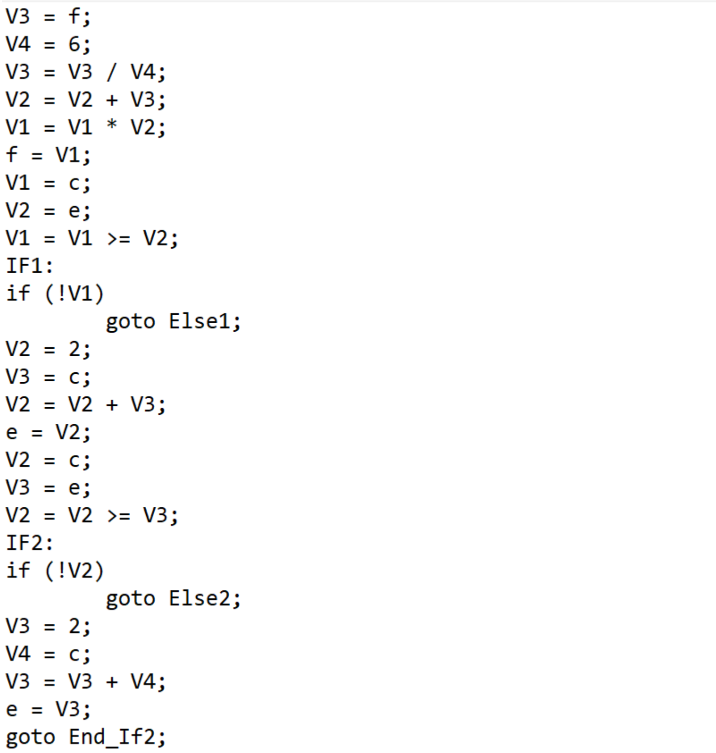


```
Error Semantico en la linea 4 tipo de dato invalido.
```

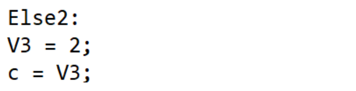
## ***Generación de código intermedio***



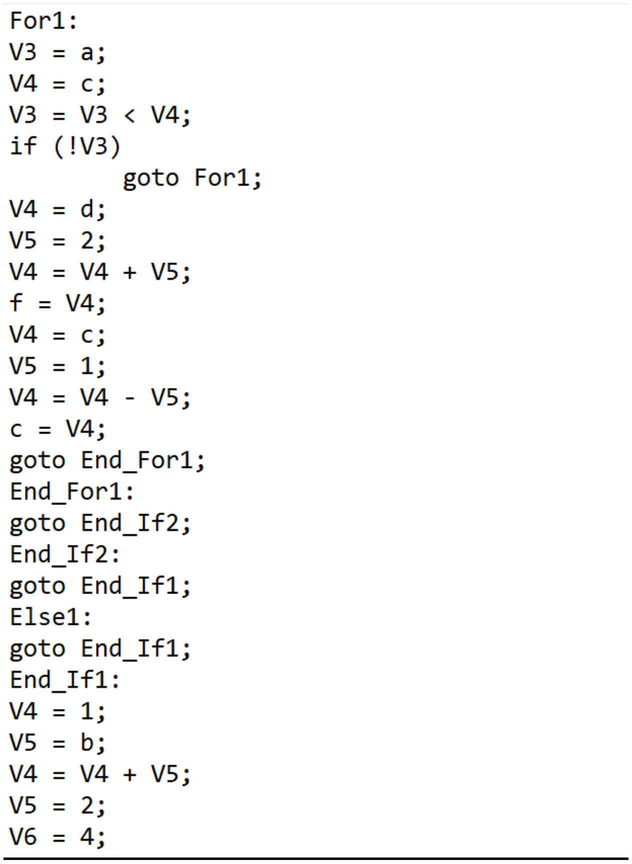
```
#include <stdio.h>

void main()
{
int a;
int b;
int c;
int gg;
float d;
float e;
float f;
float V1;
float V2;
float V3;
float V4;
float V5;
float V6;
float V7;
float V8;
float V9;
float V10;
V1 = 1;
V2 = b;
V1 = V1 + V2;
V2 = 2;
V3 = 4;
V2 = V2 * V3;
```



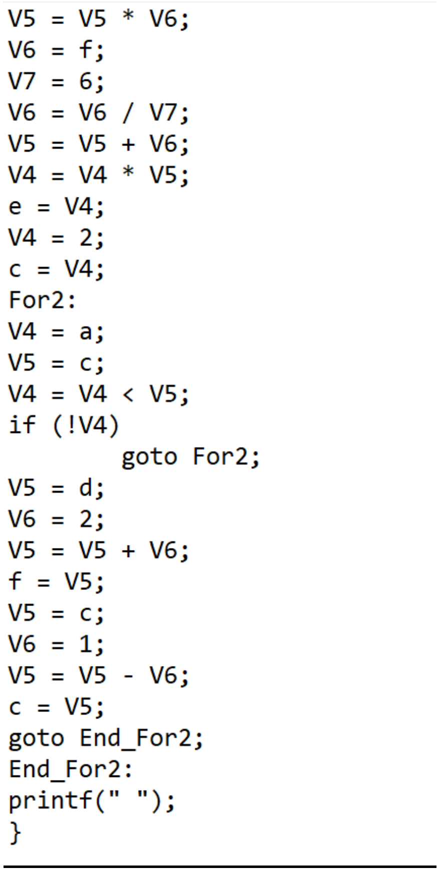
```
V3 = f;
V4 = 6;
V3 = V3 / V4;
V2 = V2 + V3;
V1 = V1 * V2;
f = V1;
V1 = c;
V2 = e;
V1 = V1 >= V2;
IF1:
if (!V1)
        goto Else1;
V2 = 2;
V3 = c;
V2 = V2 + V3;
e = V2;
V2 = c;
V3 = e;
V2 = V2 >= V3;
IF2:
if (!V2)
        goto Else2;
V3 = 2;
V4 = c;
V3 = V3 + V4;
e = V3;
goto End_If2;
```



```
Else2:
V3 = 2;
c = V3;
```

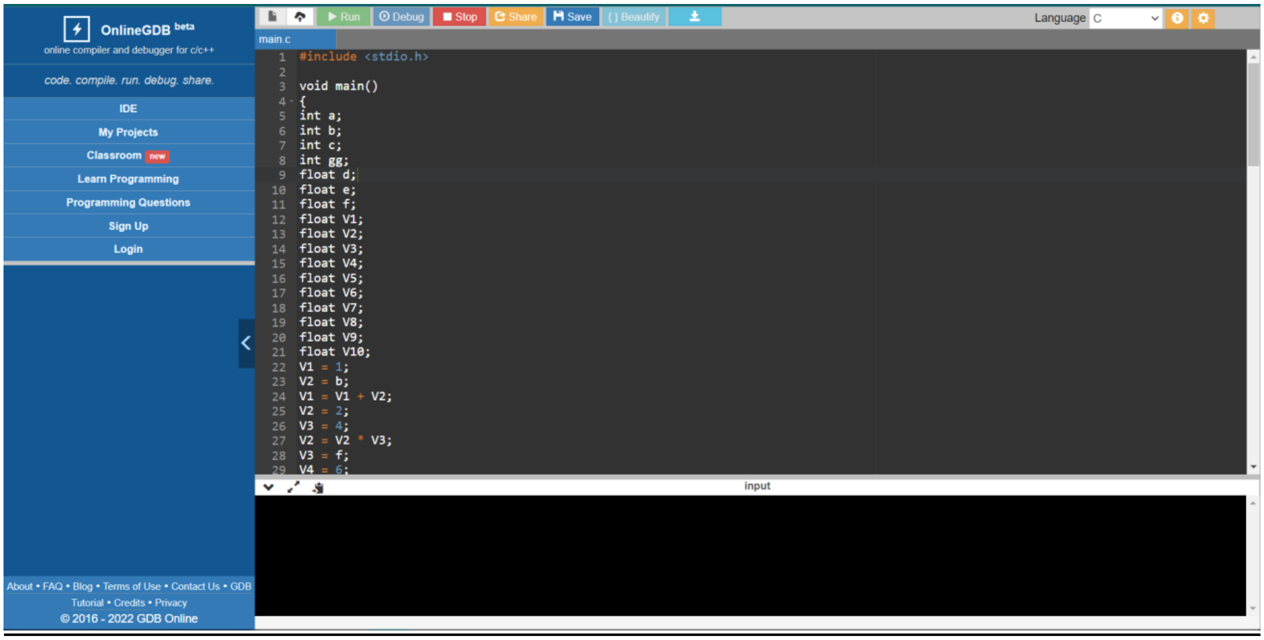


```
For1:
V3 = a;
V4 = c;
V3 = V3 < V4;
if (!V3)
        goto For1;
V4 = d;
V5 = 2;
V4 = V4 + V5;
f = V4;
V4 = c;
V5 = 1;
V4 = V4 - V5;
c = V4;
goto End_For1;
End_For1:
goto End_If2;
End_If2:
goto End_If1;
Else1:
goto End_If1;
End_If1:
V4 = 1;
V5 = b;
V4 = V4 + V5;
V5 = 2;
V6 = 4;
```



```
V5 = V5 * V6;
V6 = f;
V7 = 6;
V6 = V6 / V7;
V5 = V5 + V6;
V4 = V4 * V5;
e = V4;
V4 = 2;
c = V4;
For2:
V4 = a;
V5 = c;
V4 = V4 < V5;
if (!V4)
        goto For2;
V5 = d;
V6 = 2;
V5 = V5 + V6;
f = V5;
V5 = c;
V6 = 1;
V5 = V5 - V6;
c = V5;
goto End_For2;
End_For2:
printf(" ");
}
```

## ***Aprobación de código en la plataforma de C***



```c
#include <stdio.h>

void main()
{
int a;
int b;
int c;
int gg;
float d;
float e;
float f;
float V1;
float V2;
float V3;
float V4;
float V5;
float V6;
float V7;
float V8;
float V9;
float V10;
V1 = 1;
V2 = b;
V1 = V1 + V2;
V2 = 2;
V3 = 4;
V2 = V2 * V3;
V3 = f;
V4 = 6;
```