

```

%{
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char lexema[60];
void yyerror(char *msg);
typedef struct {
    char nombre[60];
    double valor;
    int token;
} tipoTS;
tipoTS TablaSim[100];
int nSim = 0;
int localizaSimb(char *, int);
void imprimeTablaSim();

int yylex();
int EsPalabraReservada(char[], int);
%}

%token PROGRAMA ID INICIO FIN NUM VARIABLE

%%
S: PROGRAMA ID ';' INICIO listaInstr FIN '.';
listaInstr: instr listaInstr
           | ;
instr: ID {$$ = localizaSimb(lexema,ID);} ':' '=' expr';'
expr: expr '+' term;
      | expr '-' term;
      | term;
term: term '*' term2
      | term '/' term2
      | term2;
term2: NUM {$$ = localizaSimb(lexema,NUM);}
       | ID {$$ = localizaSimb(lexema,ID)};
%%

int localizaSimb(char *nom, int tok) {
    int i;
    for(i = 0; i < nSim; i++) {
        if(!strcasecmp(TablaSim[i].nombre, nom))
            return i;
    }
    strcpy(TablaSim[nSim].nombre, nom);
    TablaSim[nSim].token = tok;
}

```

```

    if(tok == ID) TablaSim[nSim].valor = 0.0;
    if(tok == NUM) sscanf(nom, "%lf", &TablaSim[nSim].valor);
    nSim++;
    return nSim - 1;
}

void imprimeTablaSim(){
    int i;
    for(i = 0; i < nSim; i++) {
        printf("%d  nombre=%s tok=%d valor=%lf\n", i, TablaSim[i].nombre,
TablaSim[i].token, TablaSim[i].valor);
    }
}

void yyerror(char *msg){
    printf("ERROR:%s\n",msg);
}

int EsPalabraReservada(char lexema[], int default_token) {
    //strcmp considera may y minusc
    //strcasecmp ignora may de min
    if(strcasecmp(lexema,"Program")==0) return PROGRAMA;
    if(strcasecmp(lexema,"Begin")==0) return INICIO;
    if(strcasecmp(lexema,"End")==0) return FIN;
    if(strcasecmp(lexema,"Var")==0) return VARIABLE;

    return default_token;
}

int yylex(){
    char c;
    int i;
    while(1) {
        c = getchar();

        if(c == ' ') continue;
        if(c == '\t') continue;
        if(c == '\n') continue;

        if(isdigit(c)) {
            i = 0;
            do{
                lexema[i++] = c;
                c = getchar();
            } while(isdigit(c));
            ungetc(c, stdin);

```

```

        lexema[i] = '\0';
        return NUM;
    }

    if(isalpha(c)){
        i = 0;
        do{
            lexema[i++] = c;
            c = getchar();
        } while(isalnum(c));
        ungetc(c, stdin);
        lexema[i] = '\0';
        return EsPalabraReservada(lexema, ID);
    }

    return c;
}

int main(){
    if(!yyparse()) printf("La cadena es valida\n");
    else printf("La cadena es invalida\n");
    printf("tabla de simbolos\n");
    imprimeTablaSim();
    return 0;
}

```

```

[wensespl@LAPTOP-1D2RDU00 ~/USUARIO/Desktop/Compiladores/semana9]
$ ./a.out
Program MiProg;
Begin
    x:=3;
    y:=6+x;
    z:=3-4*y/2;
End.
La cadena es valida
tabla de simbolos
0  nombre=x tok=259 valor=0.000000
1  nombre=3 tok=262 valor=3.000000
2  nombre=y tok=259 valor=0.000000
3  nombre=6 tok=262 valor=6.000000
4  nombre=z tok=259 valor=0.000000
5  nombre=4 tok=262 valor=4.000000
6  nombre=2 tok=262 valor=2.000000

```

```
C:\Users\USUARIO\Desktop\Compiladores\semana9>a.exe < input.txt
```

```
La cadena es valida
```

```
tabla de simbolos
```

```
0 nombre=x tok=259 valor=0.000000
```

```
1 nombre=3 tok=262 valor=3.000000
```

```
2 nombre=y tok=259 valor=0.000000
```

```
3 nombre=6 tok=262 valor=6.000000
```

```
4 nombre=z tok=259 valor=0.000000
```

```
5 nombre=4 tok=262 valor=4.000000
```

```
6 nombre=2 tok=262 valor=2.000000
```