```
%{
#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;

typedef struct {
  int op;
  int a1;
  int a2;
  int a3;
} tipoCodigo;
int cx = -1;
tipoCodigo TCodigo[100];
void generaCodigo(int, int, int, int);

int localizaSimb(char *, int);
void imprimeTablaSim();

void imprimeTablaCod();
int nVarTemp = 0;
int GenVarTemp();

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

%token PROGRAMA ID INICIO FIN NUM VARIABLE ASIGNAR SUMAR RESTAR
MULTIPLICAR DIVIDIR PARENTESIS

%%
S: PROGRAMA ID ';' listadeclaracion INICIO listaInstr FIN '.';
listadeclaracion:  ;
listaInstr: instr listaInstr
          |  ;
```

```
instr: ID {$$ = localizaSimb(lexema, ID);} ':' '=' expr
{generaCodigo(ASIGNAR, $2, $5, '-');} ';';
expr: expr '+' term {int i = GenVarTemp(); generaCodigo(SUMAR, i, $1,
$3); $$ = i;}
    | expr '-' term {int i = GenVarTemp(); generaCodigo(RESTAR, i, $1,
$3); $$ = i;}
    | term;
term: term '*' term2 {int i = GenVarTemp(); generaCodigo(MULTIPLICAR,
i, $1, $3); $$ = i;}
    | term '/' term2 {int i = GenVarTemp(); generaCodigo(DIVIDIR, i,
$1, $3); $$ = i;}
    | term2;
term2: '(' expr ')' {int i = GenVarTemp(); generaCodigo(PARENTESIS, i,
$2, '-'); $$ = i;};
     | NUM {$$ = localizaSimb(lexema, NUM);}
     | ID {$$ = localizaSimb(lexema, ID);};
%%

int GenVarTemp(){
  char t[60];
  sprintf(t, "_T%d", nVarTemp++);
  return localizaSimb(t, ID);
}

void generaCodigo(int op, int a1, int a2, int a3){
  cx++;
  TCodigo[cx].op = op;
  TCodigo[cx].a1 = a1;
  TCodigo[cx].a2 = a2;
  TCodigo[cx].a3 = a3;
}

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 imprimeTablaCod(){
  int i;
  for(i = 0; i <= cx; i++){
    printf("%d  a1=%d a2=%d a3=%d\n", TCodigo[i].op, TCodigo[i].a1,
TCodigo[i].a2, TCodigo[i].a3);
  }
}

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();
  printf("tabla de codigos\n");
  imprimeTablaCod();
  return 0;
}
```

```
C:\Users\USUARIO\Desktop\Compiladores\semana10>a.exe < input.txt
La cadena es valida
tabla de simbolos
0  nombre=x tok=259 valor=0.000000
1  nombre=y tok=259 valor=0.000000
2  nombre=2 tok=262 valor=2.000000
3  nombre=4 tok=262 valor=4.000000
4  nombre=_T0 tok=259 valor=0.000000
5  nombre=3 tok=262 valor=3.000000
6  nombre=_T1 tok=259 valor=0.000000
7  nombre=_T2 tok=259 valor=0.000000
8  nombre=_T3 tok=259 valor=0.000000
9  nombre=5 tok=262 valor=5.000000
10  nombre=_T4 tok=259 valor=0.000000
11  nombre=_T5 tok=259 valor=0.000000
tabla de codigos
264  a1=0 a2=1 a3=45
265  a1=4 a2=2 a3=3
267  a1=6 a2=5 a3=2
269  a1=7 a2=6 a3=45
266  a1=8 a2=4 a3=7
268  a1=10 a2=9 a3=5
266  a1=11 a2=8 a3=10
264  a1=0 a2=11 a3=45
```

0 > input.txt

```
Program MiProg;
Begin
  x:=y;
  x:=2+4-(3*2)-5/3;
End.
```