%{

#include <ctype.h>

#include <stdio.h>

char lexema[255];

void yyerror(char \*);

int yylex();

%}

%token ID NUM

%%

expr: ID ':' '=' term expr

    | %empty;

term: term '+' NUM

    | term '-' NUM

    | NUM;

%%

void yyerror(char \*mgs)

{

  printf("error: %s",mgs);

}

int yylex()

{

  char c;

  while(1) {

    c = getchar();

    if(c == '\n') continue;

    if(c == ' ') continue;

    if(isspace(c)) continue;

    if(isalpha(c)) {

      int i = 0;

      do{

        lexema[i++] = c;

        c = getchar();

      } while(isalnum(c));

      ungetc(c, stdin);

      lexema[i] = 0;

      return ID;

    }

    if(isdigit(c)) {

      int i = 0;

      do{

        lexema[i++] = c;

        c = getchar();

      } while(isdigit(c));

      ungetc(c, stdin);

      lexema[i] = 0;

      return NUM;

    }

    return c;

  }

}

int main()

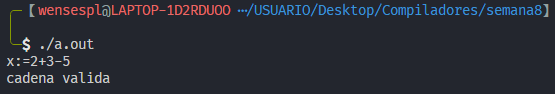
{

  if(!yyparse()) printf("cadena valida\n");

  else printf("cadena invalida\n");

  return 0;

}



|  |  |
| --- | --- |
| Estados | Cadena |
|  | X:=2+3-5 |
| 0 (ID shift, and go to state 1) | ID ‘:’ ‘=’ NUM ‘+’ NUM ‘-’ NUM |
| 01 (':' shift, and go to state 3) | ‘:’ ‘=’ NUM ‘+’ NUM ‘-’ NUM |
| 013 ('=' shift, and go to state 5) | ‘=’ NUM ‘+’ NUM ‘-’ NUM |
| 0135 (NUM shift, and go to state 6) | NUM ‘+’ NUM ‘-’ NUM |
| 01356 (reduce using rule 5 (term)) | ‘+’ NUM ‘-’ NUM |
| 0135 (term go to state 7) | ‘+’ NUM ‘-’ NUM |
| 01357 ('+' shift, and go to state 8) | ‘+’ NUM ‘-’ NUM |
| 013578 (NUM shift, and go to state 11) | NUM ‘-’ NUM |
| 013578 11 (reduce using rule 3 (term)) | ‘-’ NUM |
| 0135 (term go to state 7) | ‘-’ NUM |
| 01357 ('-' shift, and go to state 9) | ‘-’ NUM |
| 013579 (NUM shift, and go to state 12) | NUM |
| 013579 12 (reduce using rule 4 (term)) |  |
| 0135 (term go to state 7) |  |
| 01357 (reduce using rule 2 (expr)) (expr go to state 10) |  |
| 01357 10 (reduce using rule 1 (expr)) |  |
| 0 (expr go to state 2) |  |
| 02 (shift, and go to state 4) |  |
| 024 | accept |

|  |  |  |  |
| --- | --- | --- | --- |
| Estado | Stack | Entrada | Acción |
|  |  | X:=2+3-5 |  |
| 0 |  | ID ‘:’ ‘=’ NUM ‘+’ NUM ‘-’ NUM | ID shift, and go to state 1 |
| 01 | ID | ‘:’ ‘=’ NUM ‘+’ NUM ‘-’ NUM | ':' shift, and go to state 3 |
| 013 | ID ‘:’ | ‘=’ NUM ‘+’ NUM ‘-’ NUM | '=' shift, and go to state 5 |
| 0135 | ID ‘:’ ‘=’ | NUM ‘+’ NUM ‘-’ NUM | NUM shift, and go to state 6 |
| 01356 | ID ‘:’ ‘=’ NUM | ‘+’ NUM ‘-’ NUM | reduce using rule 5 (term) |
| 0135 | ID ‘:’ ‘=’ term | ‘+’ NUM ‘-’ NUM | term go to state 7 |
| 01357 | ID ‘:’ ‘=’ term | ‘+’ NUM ‘-’ NUM | '+' shift, and go to state 8 |
| 013578 | ID ‘:’ ‘=’ term ‘+’ | NUM ‘-’ NUM | NUM shift, and go to state 11 |
| 013578 11 | ID ‘:’ ‘=’ term ‘+’ NUM | ‘-’ NUM | reduce using rule 3 (term) |
| 0135 | ID ‘:’ ‘=’ term | ‘-’ NUM | term go to state 7 |
| 01357 | ID ‘:’ ‘=’ term | ‘-’ NUM | '-' shift, and go to state 9 |
| 013579 | ID ‘:’ ‘=’ term ‘-’ | NUM | NUM shift, and go to state 12 |
| 013579 12 | ID ‘:’ ‘=’ term ‘-’ NUM | $ | reduce using rule 4 (term) |
| 0135 | ID ‘:’ ‘=’ term | $ | term go to state 7 |
| 01357 | ID ‘:’ ‘=’ term | $ | reduce using rule 2 (expr) |
| 01357 | ID ‘:’ ‘=’ term expr | $ | expr go to state 10 |
| 01357 10 | ID ‘:’ ‘=’ term expr | $ | reduce using rule 1 (expr) |
| 0 | expr | $ | expr go to state 2 |
| 02 | expr $ |  | shift, and go to state 4 |
| 024 | acept |  |  |