```
%{
    #include <stdio.h>
    void yyerror(char *);
    int yylex(void);

    int sym[26];
%}

%token INTEGER VARIABLE
%left '+' '-' '*'

%%

program:
        program statement '\n'
        | /* NULL */
        ;

statement:
        expression                      { printf("%d\n", $1); }
        | VARIABLE '=' expression       { sym[$1] = $3; }
        ;

expression:
        INTEGER
        | VARIABLE                      { $$ = sym[$1]; }
        | '-' expression                { $$ = -$2; }
        | expression '+' expression     { $$ = $1 + $3; }
        | expression '-' expression     { $$ = $1 - $3; }
        | expression '*' expression     { $$ = $1 * $3; }
        | '(' expression ')'            { $$ = $2; }
        ;

%%

void yyerror(char *s) {
    fprintf(stderr, "%s\n", s);
}

int main(void) {
    yyparse();
}
```

Instructions: Run the following commands in the terminal:

```
> bison -y -d calculator.y                  <-- produces y.tab.c
> flex calculator.l                         <-- produces lex.yy.c
> gcc y.tab.c lex.yy.c                      <-- produces executable a.out

> ./a.out
a=3
(7*a)*(-3+20-15)
42
```

```
    /* calculator */
%{
    #include "y.tab.h"
    #include <stdlib.h>
    void yyerror(char *);
%}

%%

[a-z]       {
                yylval = *yytext - 'a';
                return VARIABLE;
            }

[0-9]+      {
                yylval = atoi(yytext);
                return INTEGER;
            }

[-+()=*\n]     { return *yytext; }

[ \t]   ;       /* skip whitespace */

.               yyerror("Unknown character");

%%

int yywrap(void) {
    return 1;
}
```