CS39003 Compilers Laboratory
Autumn 2026
Assignment 1
Date of posting: 21-Jun-2026

A simple interpreter on matrix operations

This assignment deals with matrices (not necessarily square and may be of various dimensions). You need to write an interpreter. The input to the interpreter is a matrix program that resides in a text file. When you feed that program file to your interpreter, it will interpret the program line by line, and take appropriate actions. You can also run the interpreter interactively. Each line in the program consists of the assignment of a matrix variable (called the l-value). The r-value (the right side of the assignment statement) can be either a literal matrix or a predefined matrix variable or the result of a binary operation on two predefined matrix variables. Use the following grammar. The terminal symbols are highlighted in red. A line **exit** terminates the interpreter (irrespective of whatever follows that line).

PROG	⟶	LINE | PROG **\n** LINE
LINE	⟶	LVAL **=** RVAL | **exit**
LVAL	⟶	**ID**
RVAL	⟶	**ID** | MATRIX | MATOP
MATRIX	⟶	**[** ROWLIST **]**
ROWLIST	⟶	ROW | ROWLIST **;** ROW
ROW	⟶	**NUM** | ROW **,** **NUM**
MATOP	⟶	**ID** OP **ID**
OP	⟶	**+** | **–** | *****

Use C-style variable names as **ID**. In this assignment, use integer values (with an optional plus or minus sign) as **NUM**. White spaces (spaces and tabs, but not new lines) are allowed anywhere except inside a variable name or a number. A line containing nothing (or only white-space character) is to be ignored.

As an example, the following line is an assignment of a 3 × 4 matrix to a matrix variable `M`. As per our grammar, commas and semicolons act as separators (not delimiters).

```
M = [ 1 , 2 , 3 , 4 ;    -5 , +6 , -7 , 8 ;       009 ,  -010 , +011 , -012 ]
```

Here are some examples of matrix operations and matrix copy. Note that the three names (the two operands and the result) are not needed to be different.

```
B1_B2 = B1 - B2
D=C-D
A   =   B
```

For every matrix assignment, the result is stored against the l-value in a symbol table. A matrix operation is permitted if and only if the two operands are of compatible dimensions. In case they are not, the l-value will store the empty (that is, the 0 × 0) matrix. Assume that the user enters the literal matrices in the correct format. In particular, the numbers of elements in all the rows in the same matrix are the same. Assume also that a variable name can be of length at most 15, the maximum dimension (row or column) of a matrix is 16, and the maximum number of matrix variables that a program can deal with is 256.

Write a C/C++ program to implement the matrix-program interpreter. Develop your code from the scratch, that is, you must **not** use any compiler-specific tools (like lex or yacc).

Submit a single C/C++ file (your interpreter program) to the submission server.

Sample Output

The input file is as below.

```
M1 = [ 1 , 2 , 3, 4 ;       -5, -6, -7, -8 ;    +009, -010, +011, -012 ]
M2=[-1,2,-3;4,-5,6;-7,8,-9;10,-11,12]
M3 = M1 * M2
     M4 = M2*M1
M5 = M4
M5=M4+M5
M6=M4-M5
M6=M6+M4
M7 = [ 1, 2; 3, 4; a, b ]
M8 = M6 * M7
M9 = M7 + M8
M9 = M1 - M0

exit
```

The output of the interpreter is as follows.

```
$ M1 = [ 1 , 2 , 3, 4 ;     -5, -6, -7, -8 ;    +009, -010, +011, -012 ]
M1  =          1       2       3       4
              -5      -6      -7      -8
               9     -10      11     -12
$ M2=[-1,2,-3;4,-5,6;-7,8,-9;10,-11,12]
M2  =         -1       2      -3
               4      -5       6
              -7       8      -9
              10     -11      12
$ M3 = M1 * M2
M3  =         26     -28      30
             -50      52     -54
            -246     288    -330
$      M4 = M2*M1
M4  =        -38      16     -50      16
              83     -22     113     -16
            -128      28    -176      16
             173     -34     239     -16
$ M5 = M4
M5  =        -38      16     -50      16
              83     -22     113     -16
            -128      28    -176      16
             173     -34     239     -16
$ M5=M4+M5
M5  =        -76      32    -100      32
             166     -44     226     -32
            -256      56    -352      32
             346     -68     478     -32
$ M6=M4-M5
M6  =         38     -16      50     -16
             -83      22    -113      16
             128     -28     176     -16
            -173      34    -239      16
$ M6=M6+M4
M6  =          0       0       0       0
               0       0       0       0
               0       0       0       0
               0       0       0       0
$ M7 = [ 1, 2; 3, 4; a, b ]
*** Invalid input matrix
M7  =   EMPTY
$ M8 = M6 * M7
*** Incompatible matrices for multiplication
M8  =   EMPTY
$ M9 = M7 + M8
M9  =   EMPTY
$ M9 = M1 - M0
*** Bad variable "M0"
M9  =   EMPTY
$
$ exit
```