Read the documents on the class web site titled:
   Program Assignment Rules,
   How To Make A Test Script,
   sampleProgramSubmission-level-01,
   sampleProgramSubmission-level-02,
   sampleProgramSubmission-level-03,
   sampleTestScript-level-01,
   sampleTestScript-level-02,
   sampleTestScript-level-03
before beginning this programming assignment.

DUE DATES:

September 26: Program #1 second-level program due.
Turn in both source file and test script via email by midnight.

October 3: Program #1 final-level program due.
Turn in both source file and test script via email by midnight.

Write a program that can input 2x2 matrices with real coefficients, perform some calculations on
them, and print out the results. Write the program so that it works with lists of commands such as
the following:

----- start of sample command list -----
A = 3 4 1.24 8.39
B = 1 7 8 3
C = A + B
D = Inv(B)
Print D
E = D * B
Print E
F = 1 2 2 3.99999999
G = Inv(F)
$ 
----- end of sample command list -----
A = 3.000000 4.000000 1.240000 8.390000
An assignment command has been processed.

B = 1.000000 7.000000 8.000000 3.000000
An assignment command has been processed.

C = A + B
An assignment command has been processed.

D = Inv(B)
An assignment command has been processed.

Print D
Matrix D is:

-0.056604 0.132075
0.150943 -0.018868

E = D * B
An assignment command has been processed.

Print E
Matrix E is:

1.000000 0.000000
0.000000 1.000000

F = 1.000000 2.000000 2.000000 4.000000
An assignment command has been processed.

G = Inv(F)
Determinant is smaller than 0.0001.
Abandoning attempt to find an inverse.
An assignment command has been processed.

$ All done!

----- end of sample output here -----
The effect of the command "A = 3 4 1.24 8.39" would be to set the variable A equal to the matrix:

\[
\begin{bmatrix}
3 & 4 \\
1.24 & 8.39
\end{bmatrix}
\]

The effect of "B = 1 7 8 3" is similar, setting B to the matrix:

\[
\begin{bmatrix}
1 & 7 \\
8 & 3
\end{bmatrix}
\]

Your program must implement 26 variables, one for each upper-case letter. The effect of the command "C = A + B" would be to set variable C equal to the sum of A and B, which is:

\[
\begin{bmatrix}
3 + 1 & 4 + 7 \\
1.24 + 8 & 8.39 + 3
\end{bmatrix}
\]

which equals:

\[
\begin{bmatrix}
4 & 11 \\
9.24 & 11.39
\end{bmatrix}
\]

The effect of the command "D = Inv(B)" would be to set variable D equal to the inverse of B, which is (approximately):

\[
\begin{bmatrix}
-0.056604 & 0.132075 \\
0.150943 & -0.018868
\end{bmatrix}
\]

The effect of the command "Print D" would be to print out the matrix above with a label. This answer is rounded to the sixth decimal place. The program must round all the matrix entries it prints to six decimal places. Also, it is important that the numbers line up nicely as shown. As you may know, the inverse of a matrix of the form:

\[
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\]

is given by the formula:

\[
\begin{bmatrix}
d/s & -b/s \\
-c/s & a/s
\end{bmatrix}
\]

where \( s = ad-bc \). The formula does not make sense if \( s=0 \). Even if \( s \) is non-zero but extremely small, the formula is useless because it is not accurate. Let's agree that your program will just print out an appropriate error message if \(|s| < 0.0001\). Make sure that you do something to prevent your
program from dividing by zero! The effect of "E = D * B" would be to set E to the product of D times B. The product of these two matrices:

\[
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\begin{pmatrix}
e & f \\
g & h
\end{pmatrix}
\]

is equal to

\[
\begin{pmatrix}
ae+bg & af+bh \\
ce+dg & cf+dh
\end{pmatrix}
\]

Be careful not to make any careless assumptions when you implement this operation. Matrix multiplication is non-commutative. That means that, in general, the product of say, A * B, is NOT equal to B * A.

In the sample input list above, the $ in the last line is the signal that the input is ended. The program you write must be capable of processing any number of commands similar to the ones shown. Addition, subtraction, multiplication and inversion of arbitrary real 2x2 matrices must be supported, as must assignment to any variable whose name is an upper-case letter. The program must read from standard input and write to standard output. Expect it to be run mostly in this manner:

a.out < fileOfCommands

or

a.out < fileOfCommands > fileOfOutputs

You may write the program so it relies on the token-spacing used in the example -- one blank between successive tokens.