Matrix Multiplication visualized as a rotation of a closed figure.

The following program demonstrates matrix multiplication. The program uses a 2x2 matrix that represents the rotation of axis rules. The students create a set of vertices for their initial. Initials must be close with no crossing segments. Thus students should be clever about picking their initial. Letters with closed loops (like A and B) are more difficult to draw.

The program is set up for 11 vertices. If you wish to create a figure with more or less vertices you must edit the following line:

For (D,1,10,1) change 10 to the appropriate number of vertices.

A sample Initial creation worksheet is included with this package.

The matrix of vertices must be set up independently of the program. Use the matrix editor for your particular calculator.
In the TI-86 version, the rotation matrix is RT,
for the TI-83 the matrix is [H],
and for the TI-89 the matrix is rt.
Remember that programs on the TI-89 are case sensitive.

Once your program is entered, and your Initial matrix is created, running the program will cause the initial to rotate. Clever students have created other images.

The TI-86 instructions also include a key-by-key breakdown of the entry process. The TI-83 and TI-89 only show the program.

If you want to save your kids the hassle of entering the program, you can create it on your calculator and then link it directly to them.

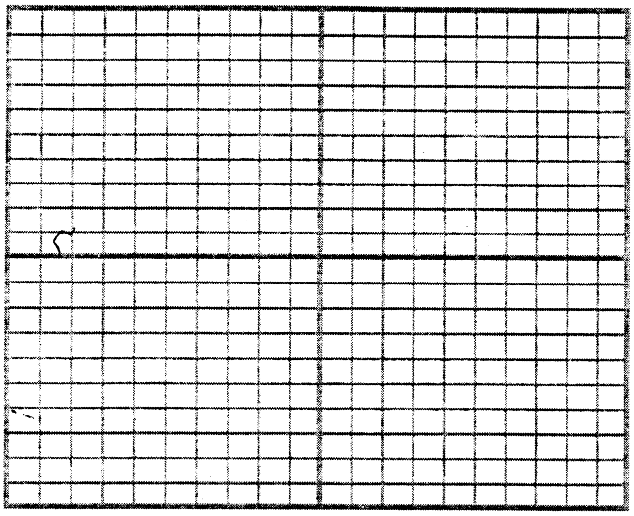
Good Luck.

Let me know if you find any typos in the programs.

Sam Koski
askoski@aol.com

Blank Initial Creation Worksheet. Be SURE your initial is centered in the first quadrant. Your initial should not touch the x or y axes.

1. Create your initial in the first quadrant.



2. List the coordinates of the endpoints of your initial

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
X																			
Y																			

8 19

3. Create a 2xN matrix to hold your endpoint data.

$\boxed{\text{2nd}}$ 7 $\boxed{\text{F2}}$ RT

```
Type in dimensions and then enter your vertex points for your your
intital.
```

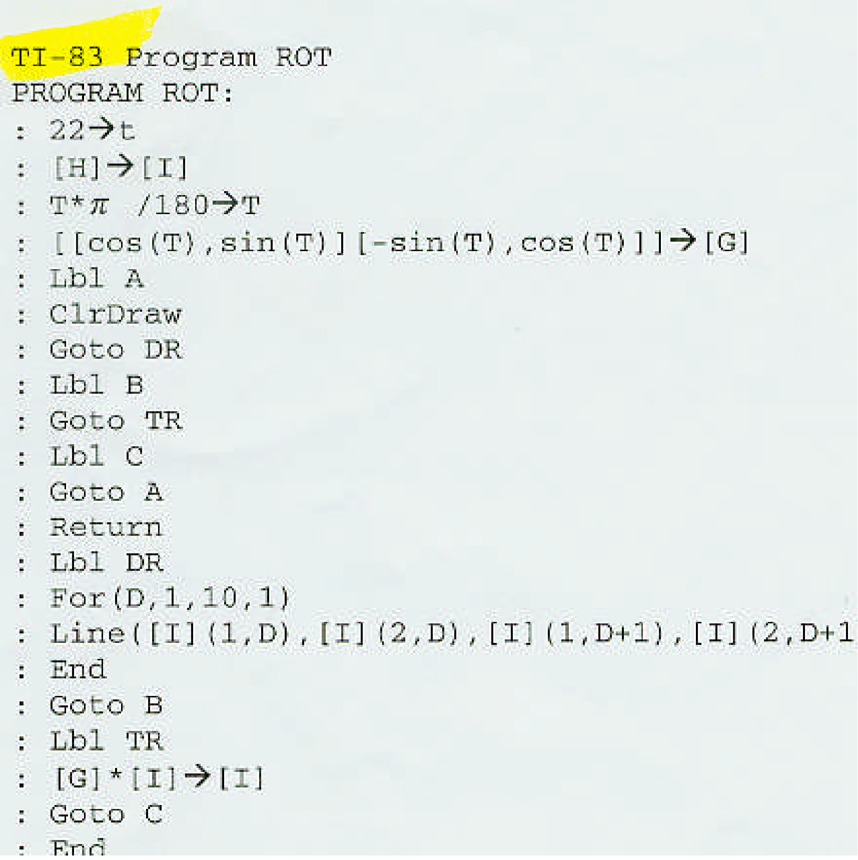
Program for the TI-86. Be sure that you put your Initials coordinates into matrix RT.

TI-86

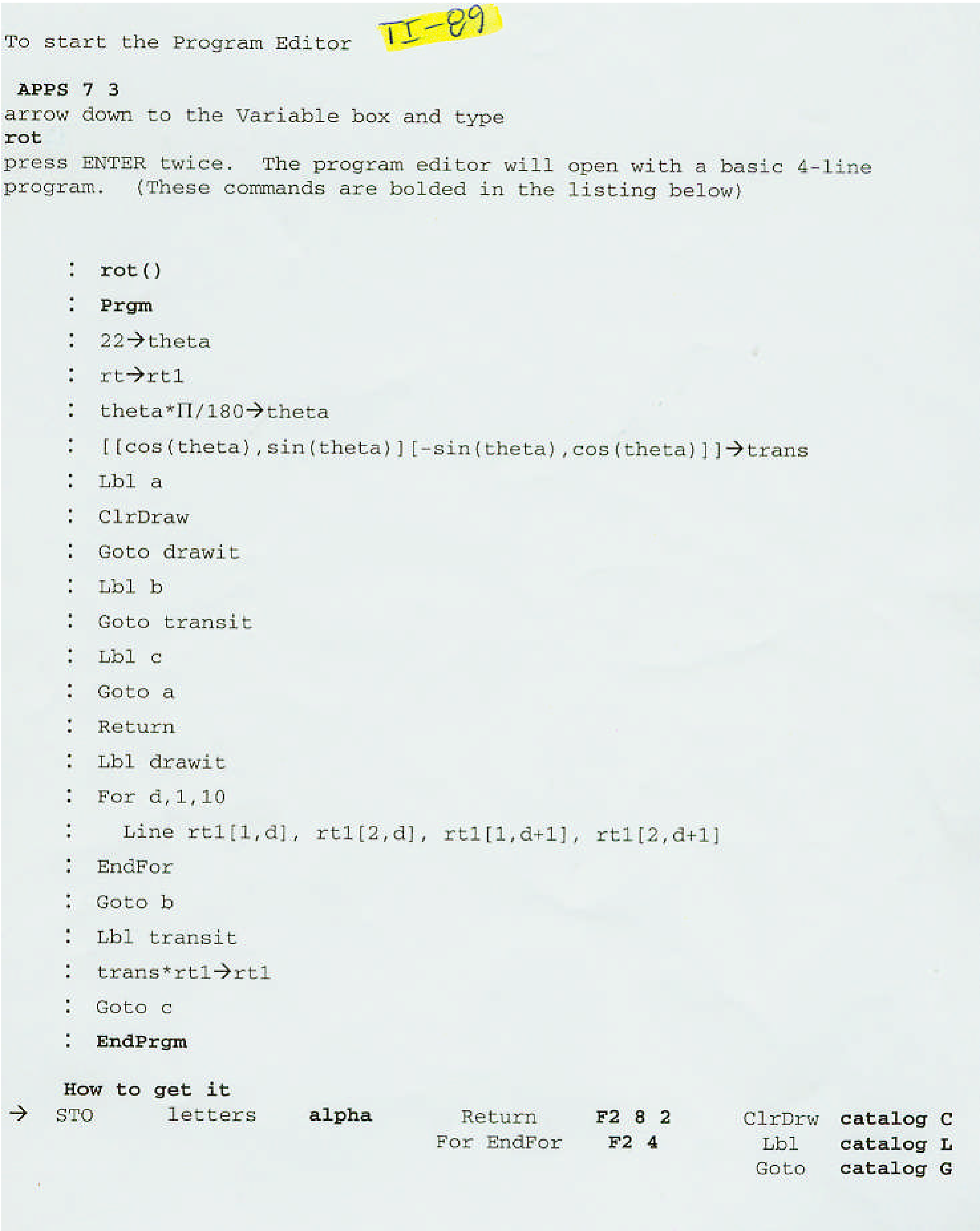
4.

```
:10→THETA
:RT→RT1
:THETA*Π/180→THETA
:[[cos (THETA),sin (THETA)][-sin (THETA),cos (THETA)]→TRANS
:Lbl A
:ClDrw
:Goto DrawIt
:Lbl B
:Goto TransIt
:Lbl C
:Goto A
:Return
:Lbl DrawIt
:For(D,1,11,1)
:Line(RT1(1,D),RT1(2,D),RT1(1,D+1),RT1(2,D+1))
:End
:Goto B
:Lbl TransIt
:TRANS*RT1→RT1
:Goto C
```

Program for the TI-83. Be sure to put your Initial matrix into [H],



```
TI-83 Program ROT
PROGRAM ROT:
: 22→t
: [H]→[I]
: T*π /180→T
: [[cos(T),sin(T)][-sin(T),cos(T)]]→[G]
: Lbl A
: ClrDraw
: Goto DR
: Lbl B
: Goto TR
: Lbl C
: Goto A
: Return
: Lbl DR
: For(D,1,10,1)
: Line([I](1,D),[I](2,D),[I](1,D+1),[I](2,D+1
: End
: Goto B
: Lbl TR
: [G]*[I]→[I]
: Goto C
: End
```



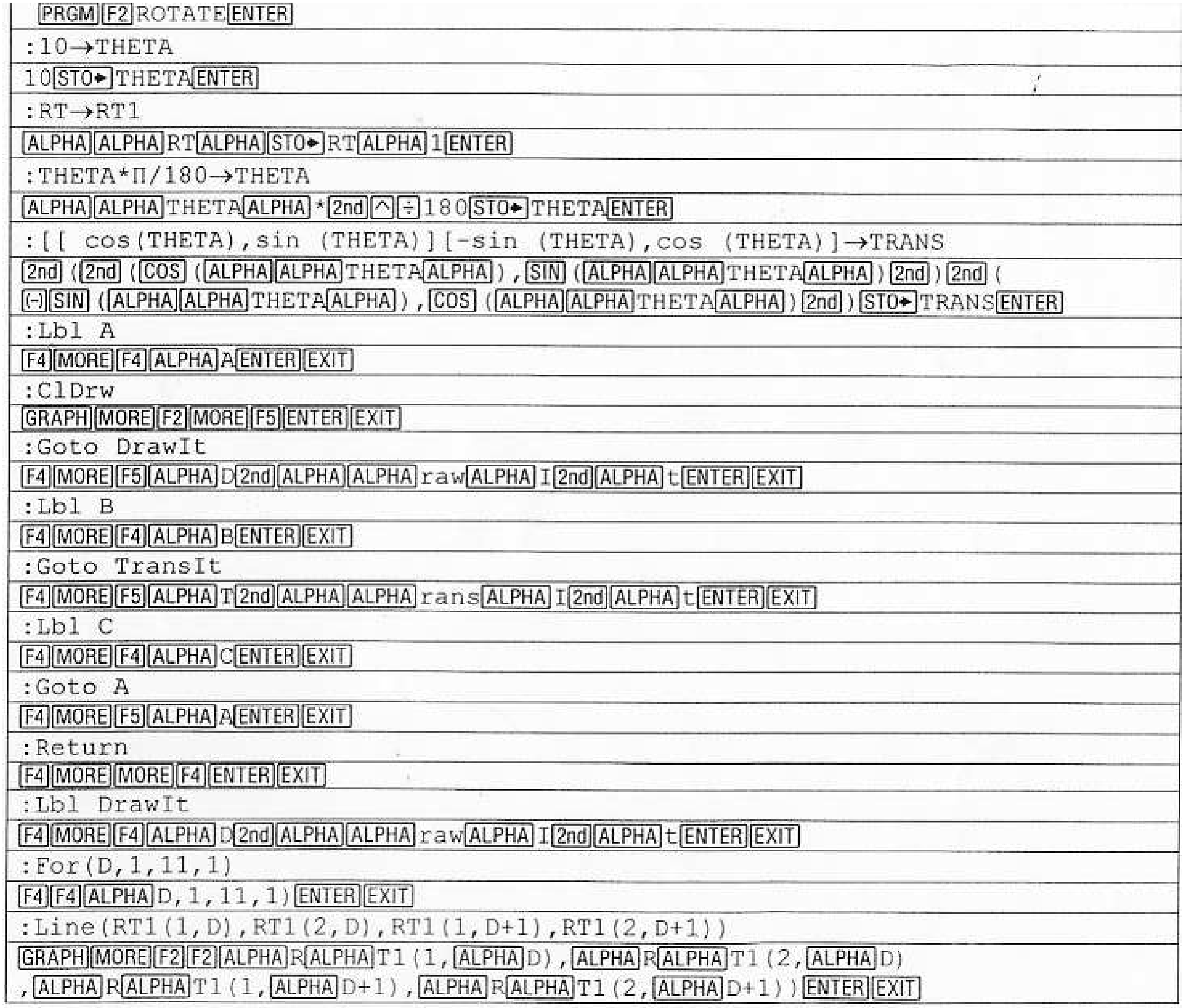
To start the Program Editor TI-89

APPS 7 3
arrow down to the Variable box and type
rot
press ENTER twice. The program editor will open with a basic 4-line program. (These commands are bolded in the listing below)

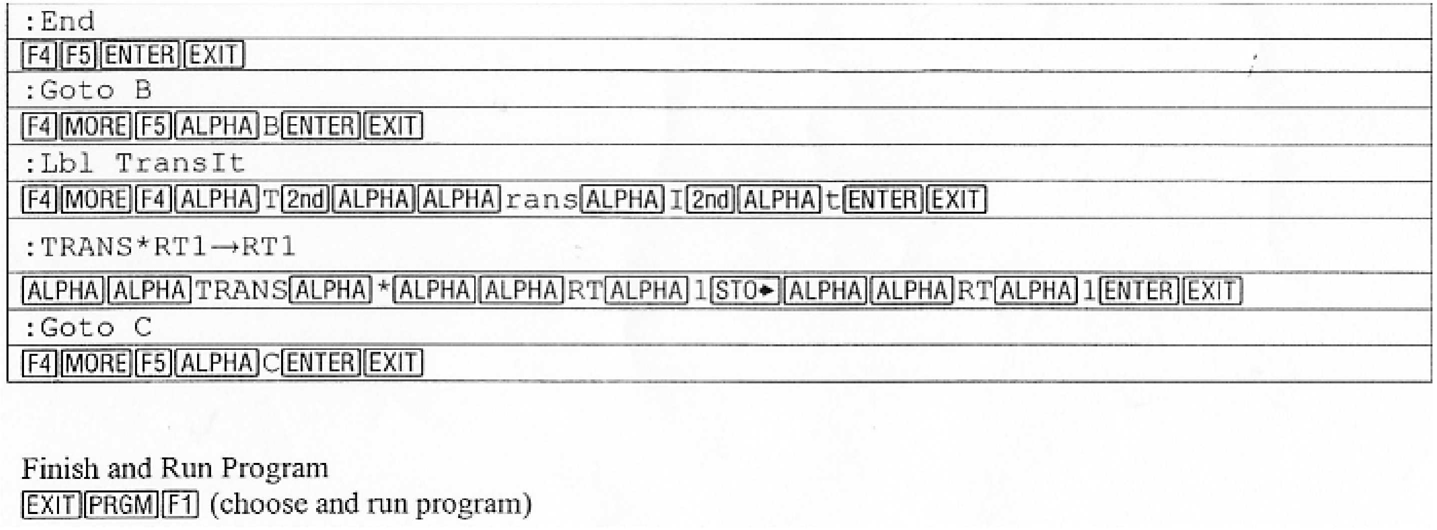
```
: rot()
: Prgm
: 22→theta
: rt→rt1
: theta*Π/180→theta
: [[cos(theta),sin(theta)][-sin(theta),cos(theta)]]→trans
: Lbl a
: ClrDraw
: Goto drawit
: Lbl b
: Goto transit
: Lbl c
: Goto a
: Return
: Lbl drawit
: For d,1,10
:   Line rt1[1,d], rt1[2,d], rt1[1,d+1], rt1[2,d+1]
: EndFor
: Goto b
: Lbl transit
: trans*rt1→rt1
: Goto c
: EndPrgm
```

How to get it

→	STO	letters	**alpha**	Return	**F2 8 2**	ClrDrw	**catalog C**
				For EndFor	**F2 4**	Lbl	**catalog L**
						Goto	**catalog G**



[PRGM][F2]ROTATE[ENTER]
:10→THETA
10[STO▸]THETA[ENTER]
:RT→RT1
[ALPHA][ALPHA]RT[ALPHA][STO▸]RT[ALPHA]1[ENTER]
:THETA*Π/180→THETA
[ALPHA][ALPHA]THETA[ALPHA]*[2nd][^][÷]180[STO▸]THETA[ENTER]
:[[cos(THETA),sin (THETA)][-sin (THETA),cos (THETA)]→TRANS
[2nd] ([2nd] ([COS] ([ALPHA][ALPHA]THETA[ALPHA]) ,[SIN] ([ALPHA][ALPHA]THETA[ALPHA]) [2nd]) [2nd] ([(-)][SIN] ([ALPHA][ALPHA]THETA[ALPHA]) ,[COS] ([ALPHA][ALPHA]THETA[ALPHA]) [2nd]) [STO▸]TRANS[ENTER]
:Lbl A
[F4][MORE][F4][ALPHA]A[ENTER][EXIT]
:ClDrw
[GRAPH][MORE][F2][MORE][F5][ENTER][EXIT]
:Goto DrawIt
[F4][MORE][F5][ALPHA]D[2nd][ALPHA][ALPHA]raw[ALPHA]I[2nd][ALPHA]t[ENTER][EXIT]
:Lbl B
[F4][MORE][F4][ALPHA]B[ENTER][EXIT]
:Goto TransIt
[F4][MORE][F5][ALPHA]T[2nd][ALPHA][ALPHA]rans[ALPHA]I[2nd][ALPHA]t[ENTER][EXIT]
:Lbl C
[F4][MORE][F4][ALPHA]C[ENTER][EXIT]
:Goto A
[F4][MORE][F5][ALPHA]A[ENTER][EXIT]
:Return
[F4][MORE][MORE][F4][ENTER][EXIT]
:Lbl DrawIt
[F4][MORE][F4][ALPHA]D[2nd][ALPHA][ALPHA]raw[ALPHA]I[2nd][ALPHA]t[ENTER][EXIT]
:For(D,1,11,1)
[F4][F4][ALPHA]D,1,11,1)[ENTER][EXIT]
:Line(RT1(1,D),RT1(2,D),RT1(1,D+1),RT1(2,D+1))
[GRAPH][MORE][F2][F2][ALPHA]R[ALPHA]T1(1,[ALPHA]D),[ALPHA]R[ALPHA]T1(2,[ALPHA]D) ,[ALPHA]R[ALPHA]T1(1,[ALPHA]D+1),[ALPHA]R[ALPHA]T1(2,[ALPHA]D+1))[ENTER][EXIT]



:End
F4 F5 ENTER EXIT
:Goto B
F4 MORE F5 ALPHA B ENTER EXIT
:Lbl TransIt
F4 MORE F4 ALPHA T 2nd ALPHA ALPHA rans ALPHA I 2nd ALPHA t ENTER EXIT
:TRANS*RT1→RT1
ALPHA ALPHA TRANS ALPHA * ALPHA ALPHA RT ALPHA 1 STO→ ALPHA ALPHA RT ALPHA 1 ENTER EXIT
:Goto C
F4 MORE F5 ALPHA C ENTER EXIT

Finish and Run Program

EXIT PRGM F1 (choose and run program)