TI-86

Quadratic Formula Program

This program will display the solutions of a quadratic equation. To use the program, write the quadratic equation in general form and enter the values of *a*, *b*, and *c*. This program gives both real and complex answers. Solutions of a quadratic equation are also available directly by using the POLY function.

PROGRAM:QUADRAT
:Disp "AX^2+BX+C=0"
:Input "ENTER A",A
:Input "ENTER B",B
:Input "ENTER C",C
:B^2−4∗A∗C→D
:(-B+√ D)/(2A)→M
:Disp M
:(-B−√ D)/(2A)→N
:Disp N

Graph Reflection Program

This program will graph a function *f* and its reflection in the line *y = x*. To use this program, enter the function in y1 and set a viewing window.

PROGRAM:REFLECT
:63∗xMin/127→yMin
:63∗xMax/127→yMax
:xScl→yScl
:y2=x
:DispG
:(xMax−xMin)/126→I
:xMin→x
:Lbl A
:PtOn(y1,x)
:x+I→x
:If x>xMax
:Stop
:Goto A

Systems of Linear Equations Program

This program will display the solution of a system of two linear equations in two variables of the form

$$\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$$

if a unique solution exists.

PROGRAM:SOLVE
:Disp "AX+BY=C"
:Input "ENTER A",A
:Input "ENTER B",B
:Input "ENTER C",C
:Disp "DX+EY=F"
:Input "ENTER D",D
:Input "ENTER E",E
:Input "ENTER F",F
:If A∗E−B∗D==0
:Goto A
:(C∗E−B∗F)/(A∗E−B∗D)→X
:(A∗F−C∗D)/(A∗E−B∗D)→Y
:Disp X
:Disp Y
:Stop
:Lbl A
:Disp "NO UNIQUE SOLUTION"

Evaluating an Algebraic Expression Program

This program can be used to evaluate an algebraic expression in one variable at several values of the variable. To use this program, enter an expression in y1. To quit the program press ON F5 .

PROGRAM:EVALUATE
:Lbl A
:Input "Enter x",x
:Disp y1
:Goto A

Fractal Fern Program

This program draws a fractal that is in the shape of a fern leaf. To use this program, enter a starting point (x, y) for the fractal. This program will take several minutes to execute. To quit the program, press $\boxed{\text{ON}}$ $\boxed{\text{F5}}$.

```
PROGRAM:FERN
:Radian
:0→xMin
:50→xMax
:10→xScl
:0→yMin
:50→yMax
:10→yScl
:ClDrw
:25→A
:25→B
:0→J
:Disp "START (X, Y)"
:Disp "ENTER X"
:Input C
:Disp "ENTER Y"
:Input D
:Lbl M
:8*(A+D)−1→U
:16*(B+C)+20→V
:PtOn(U, V)
:rand→Z
:0.85→R
:0.85→S
:-0.0436→A
:-0.0436→B
:1→K
:If Z<0.005
:Goto N
:If Z<0.1025
:Goto O
:If Z<0.2
:Goto P

:Lbl Q
:R*C*cos(A)−S*D*sin(B)→E
:R*C*sin(A)+S*D*cos(B)+K→F
:E→C
:F→D
:J+1→J
:If J<10000
:Goto M
:End
:Lbl N
:0→R
:0.16→S
:0→A
:0→B
:0→K
:Goto Q
:Lbl O
:0.3→R
:0.34→S
:0.8552→A
:0.8552→B
:1.6→K
:Goto Q
:Lbl P
:0.3→R
:0.37→S
:2.0944→A
:-0.8552→B
:0.44→K
:Goto Q
:End
```

Visualizing Row Operations Program

This program demonstrates how elementary matrix row operations used in Gauss-Jordan elimination may be interpreted graphically. It asks the user to enter a 2×3 matrix that corresponds to a system of two linear equations. (The matrix entries should not be equivalent to either vertical or horizontal lines. This demonstration is also most effective if the y-intercepts of the lines are between -10 and 10.)

   While the demonstration is running, you should notice that each elementary row operation creates an equivalent system. This equivalence is reinforced graphically because, although the equations of the lines change with each elementary row operation, the point of intersection remains the same. You may want to run this program a second time to notice the relationship between the row operations and the graphs of the lines of the system. To use this program, dimension matrix TEMP as a 2×3 matrix. Press $\boxed{\text{ENTER}}$ after each screen display to continue the program.

```
PROGRAM: ROWOPS
:Disp "ENTER A"
:Disp "2 BY 3 MATRIX:"
:Disp "A B C"
:Disp "D E F"
:Prompt A,B,C
:Prompt D,E,F
:A→TEMP(1,1):B→TEMP(1,2)
:C→TEMP(1,3):D→TEMP(2,1)
:E→TEMP(2,2):F→TEMP(2,3)
:ClLCD
:Disp "ORIGINAL MATRIX:"
:Disp TEMP
:Pause
:y2=B⁻¹(C−A*x)
:y1=E⁻¹(F−D*x)
:ZStd:Pause:ClLCD
:Disp "OBTAIN LEADING"
:Disp "1 IN ROW 1"
:multR(A⁻¹,TEMP,1)→TEMP
:Disp TEMP:Pause
:y2=(A/B)(C/A−x)
:ClDrw:DispG:Pause:ClLCD
:Disp "OBTAIN 0 BELOW"
:Disp "LEADING 1 IN"
:Disp "COLUMN 1"
:mRAdd(-D,TEMP,1,2)→TEMP
:Disp TEMP:Pause:
:y1=(E−(B*D/A))⁻¹(F−(D*C/A))
:ClDrw:DispG:Pause:ClLCD
:TEMP(2,2)→G
:If G==0
:Goto A
```

```
:multR(G⁻¹,TEMP,2)→TEMP
:Disp "OBTAIN LEADING"
:Disp "1 IN ROW 2"
:Disp TEMP:Pause:ClDrw
:DispG:Pause:ClLCD
:Disp "OBTAIN 0 ABOVE"
:Disp "LEADING 1 IN"
:Disp "COLUMN 2"
:TEMP(1,2)→H
:mRAdd(-H,TEMP,2,1)→TEMP
:Disp TEMP:Pause:FnOff 2:ClDrw
:Vert -(B/A)(E−(B*D/A))⁻¹(F−D*C/A)+C/A
:DispG:Pause:ClLCD
:Disp "THE POINT OF"
:Disp "INTERSECTION IS"
:Disp "X=",TEMP(1,3),"Y=",TEMP(2,3)
:Stop
:Lbl A
:If TEMP(2,3)==0
:Then
:Disp "INFINITELY MANY"
:Disp "SOLUTIONS"
:Else
:Disp "INCONSISTENT"
:Disp "SYSTEM"
:End
```

Graphing a Sine Function Program

This program will simultaneously draw a unit circle and the corresponding points on the sine curve. After the circle and sine curve are drawn, the points on the unit circle are connected with their corresponding points on the sine curve.

```
PROGRAM:SINESHOW
:Radian
:ClDrw:FnOff
:Param:SimulG
:-2.25→xMin
:π/2→xMax
:3→xScl
:-1.19→yMin
:1.19→yMax
:1→yScl
:0→tMin
:6.3→tMax
:0.15→tStep
:xt1=-1.25+cos t
:yt1=sin t
:xt2=t/4
       :yt2=sin t
:DispG
:For(N,1,12)
:N∗π/6.5→t
       :-1.25+cos t→A
:sin t→B
:t/4→C
:Line(A,B,C,B)
:Pause
:End
:Pause :Func
:SeqG:Disp
```

Finding the Angle Between Two Vectors Program

This program will graph two vectors and calculate the measure of the angle between the vectors. Be sure to set an appropriate viewing window. After the vectors are drawn, press ENTER to view the angle between the vectors.

```
PROGRAM:VECANGL
:ClLCD
:Degree
:Disp "ENTER (A,B)"
:Input "ENTER A",A
:Input "ENTER B",B
:ClLCD
:Disp "ENTER (C,D)"
:Input "ENTER C",C
:Input "ENTER D",D
:Line(0,0,A,B)
:Line(0,0,C,D)
:Pause
:A∗C+B∗D→E
:√ (A²+B²)→U
:√ (C²+D²)→V
:cos⁻¹(E/(U∗V))→θ
:ClDrw:ClLCD
:Disp "θ=",θ
:Stop
```

Adding Vectors Graphically Program

This program will graph two vectors in standard position. Using the parallelogram law for vector addition, the program also graphs the vector sum. Be sure to set an appropriate viewing window.

```
PROGRAM:ADDVECT
:ClDrw
:Disp "ENTER (A,B)"
:Input "ENTER A",A
:Input "ENTER B",B
:Disp "ENTER (C,D)"
:Input "ENTER C",C
:Input "ENTER D",D
:Line(0,0,A,B)
:Line(0,0,C,D)
:A+C→E
:B+D→F
:Line(0,0,E,F)
:Line(A,B,E,F)
:Line(C,D,E,F)
:Pause
:Stop
```