TI-92

Quadratic Formula Program

This program will display the solutions of a quadratic equation or the words "No Real Solution." 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.

```
:quadrat( )
:Prgm
:setMode("Complex Format","RECTANGULAR")
: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))/(2*a)→m
:(-b-√(d))/(2*a)→n
:Disp m
:Disp n
:setMode("Complex Format","REAL")
:EndPrgm
```

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 rectangle.

```
:reflect ( )
:Prgm
:103xmin/239→ymin
:103xmax/239→ymax
:xscl→yscl
:x→y2(x)
:DispG
:(xmax-xmin)/238→n
:xmin→x
:While x<xmax
:   PtOn y1(x),x
:   x+n→x
:EndWhile
:EndPrgm
```

Systems of Linear Equations Program

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

$$ax + by = c$$

$$dx + ey = f$$

if a unique solution exists.

```
:solvelin( )
:Prgm
:ClrIO
:Disp "Ax+By=C"
:Input "Enter A.",a
:Input "Enter B.",b
:Input "Enter C.",c
:ClrIO
:Disp "Dx+Ey=F"
:Input "Enter D.",d
:Input "Enter E.",e
:Input "Enter F.",f
:If a*e-d*b=0 Then
:      Disp "No unique solution"
:   Else
:      (c*e-b*f)/(a*e-d*b)→x
:      (a*f-c*d)/(a*e-d*b)→y
:      Disp x
:      Disp y
:EndIf
:EndPrgm
```

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. Press ENTER after each screen display to continue the program.

```
:rowops( )
:Prgm
:ClrIO
:ClrHome
:setMode("Split Screen","Left-Right")
:setMode("Split 1 App","Home")
:setMode("Split 2 App","Graph")
: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,b,c][d,e,f]]→mat1
:ClrIO
:b^(-1)*(c−a*x)→y2(x)
:e^(-1)*(f−d*x)→y1(x)
:ZoomStd
:Disp "ORIGINAL MATRIX:"
:Pause mat1
:ClrIO
:a/b*(c/a−x)→y2(x)
:Disp "OBTAIN LEADING"
:Disp "1 IN ROW 1"
:mRow(a^(-1),mat1,1)→mat1
:Pause mat1
:ClrIO
:(e−b*d/a)^(-1)*(f−d*c/a)→y1(x)
:DispG
:Disp "OBTAIN 0 BELOW"
:Disp "LEADING 1 IN"
:Disp "COLUMN 1"
:mRowAdd(-d,mat1,1,2)→mat1
:Pause mat1
:ClrIO
:mat1[2,2]→g
:If g=0
:Goto a1
:Disp "OBTAIN LEADING"
:Disp "1 IN ROW 2"
:mRow(g^(-1),mat1,2)→mat1
:Pause mat1
:ClrIO
:mat1[1,2]→h
:FnOff 2
:LineVert -b/a*(e−b*d/a)^(-1)*(f−d*c/a)+c/a
:Disp "OBTAIN 0 ABOVE"
:Disp "LEADING 1 IN"
:Disp "COLUMN 2"
:mRowAdd(-h,mat1,2,1)→mat1
:Pause mat1
:ClrIO
:Disp "THE POINT OF"
:Disp "INTERSECTION IS"
:Disp "X=",mat1[1,3],"Y=",mat1[2,3]
:Goto A2
:Lbl a1
:If mat1[2,3]=0 Then
:Disp "INFINITELY MANY"
:Disp "SOLUTIONS"
:Else
:Disp "INCONSISTENT"
:Disp "SYSTEM"
:EndIf
:Lbl A2
:Pause
:setMode("Split Screen","Full")
:EndPrgm
```

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.

```
:evaluate( )
:Prgm
:Lbl one
:Input "enter x",x
:Disp y1(x)
:Goto one
:EndPrgm
```

Adding Vectors Graphically Program

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

```
:addvect( )
:Prgm
:ClrIO
:Input "ENTER a",a
:Input "ENTER b",b
:Input "ENTER c",c
:Input "ENTER d",d
:ClrDraw
: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
:setMode("Split 1 App","Home")
:Stop
:EndPrgm
```

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, you can connect the points on the unit circle with their corresponding points on the sine curve by pressing ENTER.

```
:sineshow( )
:Prgm
:Disp
:ClrDraw:FnOff
:setMode("Graph","Parametric")
:setGraph("Graph Order","Simul")
:-2.9→xmin
:3π/4→xmax
:3→xscl
:-1.1→ymin
:1.1→ymax
:1→yscl
:0→tmin
:6.3→tmax
:.15→tstep
:-1.25+cos(t)→xt1(t)
:sin(t)→yt1(t)
:t/4→xt2(t)
:sin(t)→yt2(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
:EndFor
:Pause
:setMode("Graph","Function")
:setGraph("Graph order","Seq")
:setMode("Split 1 App","Home")
:EndPrgm
```

Finding the Angle Between Two Vectors Program

This program will sketch two vectors and calculate the measure of the angle between the vectors. Be sure to set an appropriate viewing rectangle.

```
:vecangl( )
:Prgm
:FnOff
:ClrHome:ClrDraw
:SetMode("Split Screen","Left-Right")
:SetMode("Split 1 App","Home")
:SetMode("Split 2 App","Graph")
:SetMode("Exact/Approx","Approximate")
:ClrIO
:Disp "ENTER (A,B)"
:Input "ENTER A",A
:Input "ENTER B",B
:Line(0,0,A,B)
:Pause
:ClrIO
:Disp "ENTER (C,D)"
:Input "ENTER C",C
:Input "ENTER D",D
:Line(0,0,C,D)
:Pause
:ClrIO
:A*C+B*D→E
:√((A^2+B^2))→U
:√(C^2+D^2)→V
:cos^-1 (E/(U*V))→θ
:Disp "θ=",θ
:Pause
:SetMode("Exact/Approx","Auto")
:SetMode("Split Screen","Full")
:SetMode("Split 1 App","Home")
:Stop
:EndPrgm
```