Includes easy-entry keystroke listings.

HEWLETT-PACKARD

HP-75

USERS' LIBRARY SOLUTIONS

Math I

NOTICE

The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

HEWLETT PACKARD

HP-75 SOLUTIONS BOOK/MEDIA ORDER FORM

All HP-75 Solutions Books are available recorded from the Users' Library* or from any Hewlett-Packard Full Service Dealer. The individual Solutions Books are $12.50 each; recorded mini-data cassettes are $12.00 for each Solutions Book requested.

Telephone the Users' Library directly (503-757-2000, ext. 3371) with your Solutions Book request, your order will ship that SAME DAY; or, call our TOLL FREE number (800-547-3400, Continental U.S. only), and orders ship within 48 hours. Mail requests also ship within 48 hours of receipt.

HP-75 Users' Library Solutions Books

00075-90061	Math I
00075-90062	Math II
00075-90063	Math III
00075-90065	Games
00075-90066	Electronics
00075-90067	Statistics
00075-90070	Real Estate
00075-90071	Finance
00075-90076	Test Statistics

Solutions Book Part Number	Title	Book Only	Mini-Data Cassettes	Price
00075-90_____				
00075-90_____				
00075-90_____				
	Please add Handling Charge on orders including Solutions Books			**3.50**
			State and Local Taxes	
			TOTAL	

All payments must be in U.S. dollars and drawn on a U.S. bank.
Send orders (with payment) directly to: HEWLETT-PACKARD USERS' LIBRARY
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330

☐ Payment enclosed ☐ Master Card ☐ VISA ☐ Purchase order

Card Number/Company Purchase Order Exp

Name

Company (if applicable)

Address Telephone

City State/Country Zip

*Recorded Solutions Book programs shipped to U.S. customers only.

TABLE OF CONTENTS

PROGRAM DESCRIPTION

SIMULTANEOUS LINEAR EQUATIONS

This program solves M sets of N by N linear equations that have identical coefficients using the Crout algorithm with row interchange. If M or N is greater than 10, the dimensions of the subscripted variable in line 70 will have to be changed. The user must enter the coefficients and the constants of the equations as prompted.

If the system of equations has no solution (is linearly dependent) the program will so indicate.

A theoretical representation of a system of linear equations is:

$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + \ldots + A_{1N}x_N = B_{1j}$$

$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + \ldots + A_{2N}x_N = B_{1j}$$

$$A_{N1}x_1 + A_{N2}x_2 + A_{N3}x_3 + \ldots + A_{NN}x_N = B_{Nj}$$

Where N = number of coefficients

M = Number of sets

A_{ij} = coefficient of ith row and jth variable

B_{ij} = constant of ith row (equation) and jth set

j = 1 . . . M

There are M sets of answers :

Set 1 = $x_1, x_2, x_3, \ldots, x_N$

Set 2 = $x_1, x_2, x_3, \ldots, x_N$

Set M = $x_1, x_2, x_3, \ldots, x_N$

SAMPLE PROBLEM

Solve the following system of linear equations:

$x_1 + x_2 + x_3 = 12$ $\quad x_1 + x_2 + x_3 = 16$

$2x_1 + 3x_2 - 6x_3 = -13$ $\quad 2x_1 + 3x_2 - 6x_3 = 50$

$-x_1 + 4x_2 + 7x_3 = 43$ $\quad -x_1 + 4x_2 + 7x_3 = 26$

$M = 2$

$N = 3$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		SIMULTANEOUS LINEAR EQUATIONS	
1	Enter number of sets	Number of sets?	2 [RTN]
2	Enter number of coefficients	Number of coefficients?	3 [RTN]
	per equation		
3	Enter coefficients and	Equation #1 X#1?	1 [RTN]
	constants for each equation	Equation #1 X#2?	1 [RTN]
		Equation #1 X#3?	1 [RTN]
		Equation #1 constant #1?	12 [RTN]
		Equation #1 constant #2?	16 [RTN]
	Equation #2	Equation #2 X#1?	2 [RTN]
		Equation #2 X#2?	3 [RTN]
		Equation #2 X#3?	-6 [RTN]
		Equation #2 constant #1?	-13 [RTN]
		Equation #2 constant #2?	50 [RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
	Equation #3	Equation #3 X#1?	-1 [RTN]
		Equation #3 X#2?	4 [RTN]
		Equation #3 X#3?	7 [RTN]
		Equation #3 constant 1?	43 [RTN]
		Equation #3 constant #2?	26 [RTN]
4	Display 2 answer sets	Answer set #1	[RTN]
	Use [RTN] to view next answer,	4.00	[RTN]
	[BACK] to view previous answer	3.00	[RTN]/[BACK]
		5.00	[RTN]/[BACK]
		Answer set #2	[RTN]
		7.00	[RTN]
		10.00	[RTN]/[BACK]
		-1.00	[RTN]/[BACK]
5	End	Run again, View again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		SIMULTANEOUS LINEAR EQUATIONS	
1	Enter number of sets of equa-		
	tions	Number of sets?	M [RTN]
2	Enter number of coefficients		
	per equation (also the number		
	of equations per set)	Number of coefficients	N [RTN]
3	Enter coefficients for each	Equation #1 X#1?	A_{11} [RTN]
	equation (A_{11} A_{12} A_{13} ...A_{1N}	Equation #1 X#2?	A_{12} [RTN]
	A_{21} A_{22} A_{21} ...A_{2N}	Equation #1 X#3?	A_{13} [RTN]
	A_{N1} A_{N2} A_{N3} ...A_{NN})	Equation #N X#1?	A_{N1} [RTN]
	one at a time along with the	Equation #N X#2?	A_{N2} [RTN]
	constants after each equation	Equation #N constant #M?	B_{NM} [RTN]
4	If the matrix is singular:	Matrix of coefficients singular	[RTN]
	To run again, enter 'R', else	Run again or End?	R or E [RTN]
	enter 'E' to end program.	END OF PROGRAM	
5	Display M answer sets	Answer set #1	[RTN]
	(N answers per set)	nnn.nn	[RTN]
	Use [RTN] to view next answer,	nnn.nn	[RTN]/[BACK]
	[BACK] to view previous answer	nnn.nn	[RTN]/[BACK]
		Answer set #2	[RTN]
		nnn.nn	[RTN]
		nnn.nn	[RTN]/[BACK]
		nnn.nn	[RTN]/[BACK]
		Answer set #3	[RTN]
		Answer set #M	[RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		nnn.nn	[RTN]
		nnn.nn	[RTN]/[BACK]
		nnn.nn	[RTN]/[BACK]
6	Program options	Run again, View again, or End?R	V,E or [RTN]
	If 'R' then step 1		
	If 'V' then step 5		
	If 'E then stop.	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A(,)	Array for coefficients of equations	X	Used for comparison and storage
B(,)	Array for constants of equations. Later used for answer sets.	S	Accumulates sums of products of elements
		I,J,K	Looping and indexing
T	Temporary storage in row interchange process	M1	Loop parameter used in row interchange
Q	Key to index of largest element in equation	Q$	User interaction
N	Number of coefficients (also the number of equations)	M	Number of sets of equations

NOTES AND REFERENCES

References: SIMULTANEOUS LINEAR EQUATIONS, HP-85 Math Solutions Book, Hewlett-Packard, 1981.

Fuller, Leonard E., BASIC MATRIX THEORY, Prentiss-Hall, 1962, pp. 156-164.

PROGRAM LISTING

```
 10 ! CROUT - Solves
 20 ! simultaneous linear
 30 ! equations using Crout
 40 ! algorithm.
 50 ! Revision 11/01/82.
 60 DIM Q$[4]
 70 SHORT A(10,10),B(10,10),T,S
 80 INTEGER I,J,K,M,N,Q,M1
 90 DEF FNQ$                                  -Wait for RTN or BACK key
100 Z=NUM(KEY$) @ IF Z#13 AND Z#8 THEN
    100 ELSE FNQ$=CHR$(Z)
110 END DEF
120 DISP ' SIMULTANEOUS LINEAR EQUATION
    S' @ WAIT 2
130 INPUT 'Number of sets?';M
140 IF M<=0 THEN 130
150 INPUT 'Number of coefficients?';N
160 IF N<=0 THEN 150
170 FOR I=1 TO N
180 FOR J=1 TO N
190 DISP 'Equation#';I;'  X#';J;             -Enter coefficients for each
                                              equation
200 INPUT '?';A(I,J)
210 NEXT J
220 FOR K=1 TO M
230 DISP 'Equation#';I;'  constant#';K;      -Enter constants for each
                                              equation
240 INPUT '?';B(I,K)
250 NEXT K
260 NEXT I
270 FOR I=1 TO N
280 X=-1
290 FOR J=I TO N
300 IF ABS(A(J,I))<=X THEN 330               -Check for linear dependence
310 Q=J
320 X=ABS(A(J,I))
330 NEXT J
340 IF X>0 THEN 390                          -Check for matrix singularity
350 DISP 'Matrix of coefficients singul
    ar' @ GOSUB 930
360 DISP CHR$(210);'un again, or ';CHR$
    (197);
370 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
380 ON POS('RE',Q$)+1 GOTO 360,130,910
390 IF I=Q THEN 500                          -Begin row interchange process
400 FOR J=1 TO N
410 T=A(I,J)
420 A(I,J)=A(Q,J)
430 A(Q,J)=T
440 NEXT J
450 FOR J=1 TO M
460 T=B(I,J)
470 B(I,J)=B(Q,J)
```

PROGRAM LISTING

```
480 B(Q,J)=T
490 NEXT J
500 FOR J=1 TO N
510 IF I<J THEN 540
520 M1=J-1
530 GOTO 550
540 M1=I-1
550 S=0
560 FOR K=1 TO M1
570 S=S+A(I,K)*A(K,J)
580 NEXT K
590 A(I,J)=A(I,J)+S
600 IF I>=J THEN 620
610 A(I,J)=-A(I,J)/A(I,I)
620 NEXT J
630 NEXT I
640 FOR J=1 TO M
650 FOR I=1 TO N
660 S=0
670 FOR K=1 TO I-1
680 S=S+A(I,K)*B(K,J)
690 NEXT K
700 B(I,J)=-(B(I,J)+S)/A(I,I)
710 NEXT I
720 FOR I=N TO 1 STEP -1
730 S=0
740 FOR K=I+1 TO N
750 S=S+A(I,K)*B(K,J)
760 NEXT K
770 B(I,J)=-B(I,J)+S
780 NEXT I
790 NEXT J
800 ! OUTPUT.                                   -Display answers
810 FOR J=1 TO M
820 DISP 'Answer set #';J @ GOSUB 930
830 FOR I=1 TO N
840 DISP USING 'mdddd.dd' ; B(I,J)
850 Q$=FNQ$ @ IF NUM(Q$)=8 THEN I=MAX(1
    ,I-1) @ GOTO 840
860 NEXT I
870 NEXT J
880 DISP CHR$(210);'un again, ';CHR$(21   -Continuation options
    4);'iew again, or ';CHR$(197);
890 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
900 ON POS('RVE',Q$)+1 GOTO 880,130,800
    ,910
910 DISP '          END OF PROGRAM'
920 STOP
930 !
940 IF NUM(KEY$)#13 THEN 940                  -Wait for the RTN key to be
                                               pressed
950 RETURN
```

PROGRAM DESCRIPTION

QUADRATIC EQUATION

This program analyzes a quadratic equation of the form $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = Ø$. The user must supply the coefficients A, B, C, D, E, F, in that order. If a coefficient is equal to zero it must be entered as zero and not skipped.

The program determines lines, single points, circles, ellipses, hyperbolas, parabolas, parallel lines, and equations for which there are no real solutions.

SAMPLE PROBLEM

1) $x^2 + 16y^2 - 25 = \emptyset$

2) $x^2 + 4y^2 + 9 = \emptyset$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		QUADRATIC EQUATION	
1	Enter coefficients	Enter A,B,C,D,E,F?	1,0,16,0,0,
			-25, [RTN]
2	Display equation solution	Ellipse with eccentricity .97	[RTN]
	Use [BACK] to display the	Center = (0,0)	[RTN]/[BACK]
	previous result.	Angle = 0	[RTN]/[BACK]
		Focus = (4.84,0)	[RTN]/[BACK]
		Focus = (-4.83,0)	[RTN]/[BACK]
		Sum of radii = 10	[RTN]/[BACK]
		Major axis = 10	[RTN]/[BACK]
		Minor axis = 2.5	[RTN]/[BACK]
		Focal chord = .63	[RTN]/[BACK]
		Major 0.00x + 1.00y = 0.00	[RTN]/[BACK]
		Minor 1.00x + 0.00y = 0.00	[RTN]/[BACK]
		Dir 1.00x + 0.00y = 5.16	[RTN]/[BACK]
		Dir 1.00x + 0.00y = -5.16	[RTN]/[BACK]
		Area = 19.63	[RTN]/[BACK]
3	Run again for problem #2	Run again? Y	[RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
4	Enter coefficients	Enter A,B,C,D,E,F?	1,0,4,0,0,9
			[RTN]
5	Display	No real solution	[RTN]
6	End	Run again? Y	N [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		QUADRATIC EQUATION	
1	Enter coefficients of Ax^2 +	Enter A,B,C,D,E,F?	A,B,C,D,E,F
	Bxy + Cy^2 + Dx + Ey = 0		[RTN]
2	Determine type of equation		[RTN]/[BACK]
	compute statistics, and dis-		
	play. You may use the [RTN] or		
	[BACK] key to scroll through		
	the statistics for circle,		
	ellipse, hyperbola or		
	parabola.		
3	Program options	Run again? Y	N or [RTN]
	If Y then step 1 else	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A,B,C,D,E,F	Coefficients of quadratic equation	B1	Used to determine minor axes
X,Y	Coordinates of center of circle	C1	Computed constant
U,V	Used to determine foci	Q,S	Part of simple quadratic formula
H1,K1	Center of parabola	K,L,F1	Used to form new equation
G	Measure of angle	M,N	Used to determine center of parabola
A1	Used to determine major axes		

PROGRAM LISTING

```
 10 ! QUAD - Analyzes a
 20 ! quadratic equation.
 40 ! Revision 11/01/82.
 60 DIM Q$[3],W$[2]
 70 REAL Q,X,Y,F1,G,U,V,K,L,A1,B1,C1,M,
    N,S,K1,H1
 80 SHORT A,B,C,D,E,F
 90 DEF FNW$                          -Wait for RTN or BACK key
100 W=NUM(KEY$) @ IF W#13 AND W#8 THEN
    100
110 FNW$=CHR$(W)
120 END DEF
130 DEF FND(X) = IP(X*100+.5)/100     -Round number to two decimal
                                       places
140 IMAGE k,mdd.dd,k,mdd.dd,k,mdd.dd
150 DISP '        QUADRATIC EQUATION' @
    WAIT 2
160 ON ERROR GOTO 170
170 INPUT 'Enter A,B,C,D,E,F?';A,B,C,D,
    E,F
180 OFF ERROR
190 IF A#0 OR B#0 OR C#0 THEN 250
200 IF D#0 OR E#0 THEN 220
210 GOTO 170
220 !                                 -The equation is a straight
                                       line
230 DISP USING 140 ; 'Line ',D,'x + ',E
    ,'y = ',-F @ GOSUB 1990
240 GOTO 1950
250 Q=B^2-4*A*C
260 IF Q=0 THEN 1470
270 !
280 X=(2*C*D-B*E)/Q
290 Y=(2*A*E-B*D)/Q
300 F1=-(D*X/2+E*Y/2+F)
310 G=0
320 IF B=0 THEN 360
330 G=PI/4
340 IF A=C THEN 360
350 G=.5*ATN(B/(A-C))
360 U=COS(G)
370 V=SIN(G)
380 K=A*U^2+B*U*V+C*V^2
390 L=A*V^2-B*U*V+C*U^2
400 IF K>0 THEN 450
410 K=-K
420 L=-L
430 F1=-F1
440 !
450 IF Q>0 THEN 1000
460 IF F1>0 THEN 520
470 IF F1=0 THEN 500
480 !                                 -The equation has no real
                                       solution
```

PROGRAM LISTING

```
490 DISP 'No real solution.' @ GOSUB 19
    90 @ GOTO 1950
500 !                                    -The equation refers to a
                                          single point
510 DISP 'Single point at:';FND(X);',';
    FND(Y) @ GOSUB 1990 @ GOTO 1950
520 IF K#L THEN 630
530 !
540 !                                    -The equation represents a
                                          circle
550 DISP 'Circle with eccentricity 0.'
    @ GOSUB 1990
560 DISP 'Center = (';FND(X);',';FND(Y)
    ;')'
570 W$=FNW$ @ IF NUM(W$)=8 THEN 550
580 DISP 'Radius = ';FND(SQR(F1/K))
590 W$=FNW$ @ IF NUM(W$)=8 THEN 560
600 DISP 'Area = ';FND(PI*F1/K)
610 W$=FNW$ @ IF NUM(W$)=8 THEN 580
620 GOTO 1950
630 IF K<L THEN 670
640 G=G+PI/2
650 GOTO 360
660 !                                    -The equation represents an
                                          ellipse
670 A1=SQR(F1/ABS(K))
680 B1=SQR(F1/ABS(L))
690 C1=SQR(A1^2-B1^2)
700 !
710 DISP 'Ellipse with eccentricity';FN
    D(C1/A1) @ GOSUB 1990
720 DISP 'Center = (';FND(X);',';FND(Y)
    ;')'
730 W$=FNW$ @ IF NUM(W$)=8 THEN 710
740 DISP 'Angle =';FND(G*180/PI)
750 W$=FNW$ @ IF NUM(W$)=8 THEN 720
760 DISP 'Focus = (';FND(X+C1*U);',';FN
    D(X+C1*V);')'
770 W$=FNW$ @ IF NUM(W$)=8 THEN 740
780 DISP 'Focus = (';FND(X-C1*U);',';FN
    D(Y-C1*V);')'
790 W$=FNW$ @ IF NUM(W$)=8 THEN 760
800 DISP 'Sum of radii =';FND(2*A1)
810 W$=FNW$ @ IF NUM(W$)=8 THEN 780
820 DISP 'Major axis =';FND(2*A1)
830 W$=FNW$ @ IF NUM(W$)=8 THEN 800
840 DISP 'Minor axis =';FND(2*B1)
850 W$=FNW$ @ IF NUM(W$)=8 THEN 820
860 DISP 'Focal chord =';FND(2*B1^2/A1)
870 W$=FNW$ @ IF NUM(W$)=8 THEN 840
880 DISP USING 140 ; 'Major ',-V,'x + '
    ,U,'y = ',-V*X+U*Y
890 W$=FNW$ @ IF NUM(W$)=8 THEN 860
900 DISP USING 140 ; 'Minor ',U,'x + ',
    V,'y = ',U*X+V*Y
```

PROGRAM LISTING

```
 910 W$=FNW$ @ IF NUM(W$)=8 THEN 880
 920 DISP USING 140 ; 'Dir ',U,'x + ',V,
     'y = ',U*X+V*Y+A1^2/C1                    -Directix lines
 930 W$=FNW$ @ IF NUM(W$)=8 THEN 900
 940 DISP USING 140 ; 'Dir ',U,'x + ',V,
     'y = ',U*X+V*Y-A1^2/C1
 950 W$=FNW$ @ IF NUM(W$)=8 THEN 920
 960 DISP 'Area =';FND(PI*A1*B1)
 970 W$=FNW$ @ IF NUM(W$)=8 THEN 940
 980 GOTO 1950
 990 !
1000 IF F1>0 THEN 1120
1010 IF F1=0 THEN 1040
1020 G=G+PI/2
1030 GOTO 360
1040 B1=SQR(ABS(K))
1050 A1=SQR(ABS(L))
1060 !                                         -The equation represents two
                                                lines
1070 DISP USING 140 ; 'Line1 ',B1*U-A1*V
     ,'x + ',B1*V-A1*U,'y = ',(B1*U+A1*V
     )*X+(B1*V-A1*U)*Y
1080 GOSUB 1990
1090 DISP USING 140 ; 'Line2 ',B1*U-A1*V
     ,'x + ',B1*V+A1*U,'y = ',(B1*U-A1*V
     )*X+(B1*V+A1*U)*Y
1100 W$=FNW$ @ IF NUM(W$)=8 THEN 1070
1110 GOTO 1950
1120 !                                         -The equation represents a
                                                hyperbola
1130 A1=SQR(F1/ABS(K))
1140 B1=SQR(F1/ABS(L))
1150 C1=SQR(A1^2+B1^2)
1160 !
1170 DISP 'Hyperbola: eccentricity';FND(
     C1/A1) @ GOSUB 1990
1180 DISP 'Center= (';FND(X);',';FND(Y);
     ')'
1190 W$=FNW$ @ IF NUM(W$)=8 THEN 1170
1200 DISP 'Angle = ';FND(G*180/PI)
1210 W$=FNW$ @ IF NUM(W$)=8 THEN 1180
1220 DISP 'Focus = (';FND(X+C1*V);',';FN
     D(Y+C1*V);')'
1230 W$=FNW$ @ IF NUM(W$)=8 THEN 1200
1240 DISP 'Focus = (';FND(X-C1*U);',';FN
     D(Y-C1*V);')'
1250 W$=FNW$ @ IF NUM(W$)=8 THEN 1220
1260 DISP 'Radii difference =';FND(2*A1)
1270 W$=FNW$ @ IF NUM(W$)=8 THEN 1240
1280 DISP 'Major axis =';FND(2*A1)
1290 W$=FNW$ @ IF NUM(W$)=8 THEN 1260
1300 DISP 'Minor axis =';FND(2*B1)
1310 W$=FNW$ @ IF NUM(W$)=8 THEN 1280
1320 DISP 'Focal chord =';FND(2*B1^2/A1)
```

PROGRAM LISTING

```
1330 W$=FNW$ @ IF NUM(W$)=8 THEN 1300
1340 DISP USING 140 ; 'Major ',-V,'x + '
     ,U,'y = ',-V*X+U*Y
1350 W$=FNW$ @ IF NUM(W$)=8 THEN 1320
1360 DISP USING 140 ; 'Minor ',U,'x + ',
     V,'y = ',U*X+V*Y
1370 W$=FNW$ @ IF NUM(W$)=8 THEN 1340
1380 DISP USING 140 ; 'Dir ',U,'x + ',V,
     'y = ',U*X+V*Y+A1^2/C1                  -Directix lines
1390 W$=FNW$ @ IF NUM(W$)=8 THEN 1360
1400 DISP USING 140 ; 'Dir ',U,'x + ',V,
     'y = ',U*X+V*Y-A1^2/C1
1410 W$=FNW$ @ IF NUM(W$)=8 THEN 1380
1420 DISP USING 140 ; 'Asymp ',B1*U+A1*V    -Asymptote lines
     ,'x + ',B1*V-A1*U,'y = ',(B1*U+A1*V
     )*X+(B1*V-A1*U)*Y
1430 W$=FNW$ @ IF NUM(W$)=8 THEN 1400
1440 DISP USING 140 ; 'Asymp ',B1*U-A1*V
     ,'x + ',B1*V+A1*U,'y = ',(B1*U-A1*V
     )*X+(B1*V+A1*U)*Y
1450 W$=FNW$ @ IF NUM(W$)=8 THEN 1420
1460 GOTO 1950
1470 !
1480 G=0
1490 IF A=0 THEN 1530
1500 G=PI/2
1510 IF B=0 THEN 1530
1520 G=ATN(-2*A/B)
1530 L=A+C
1540 M=D*COS(G)+E*SIN(G)
1550 N=-D*SIN(G)+E*COS(G)
1560 IF ABS(M)<.00001 THEN 1800
1570 K1=-N/(2*L)
1580 H1=(-F+L*K1^2)/M
1590 C1=-L/(4*M)
1600 IF C1>=0 THEN 1630
1610 C1=-C1
1620 G=G+PI
1630 !
1640 !                                      -The equation represents a
                                             parabola
1650 DISP 'Parabola with eccentricity 1.
     ' @ GOSUB 1990
1660 DISP 'Center= (';FND(H1*COS(G)-K1*S
     IN(G));',';FND(H1*SIN(G)+K1*COS(G))
     ;')'
1670 W$=FNW$ @ IF NUM(W$)=8 THEN 1650
1680 DISP 'Angle=';FND(G*100/PI)
1690 W$=FNW$ @ IF NUM(W$)=8 THEN 1660
1700 DISP 'Focus= (';FND((H1+C1)*COS(G)-
     K1*SIN(G));',';
1710 DISP FND((H1+C1)*SIN(G)+K1*COS(G));
     ')'
1720 W$=FNW$ @ IF NUM(W$)=8 THEN 1680
```

PROGRAM LISTING

```
1730 DISP 'Focal chord=';FND(4*C1)
1740 W$=FNW$ @ IF NUM(W$)=8 THEN 1700
1750 DISP USING 140 ; 'Sym ',-SIN(G),'x
     + ',COS(G),'y = ',K1                    -Axis of symmetry
1760 W$=FNW$ @ IF NUM(W$)=8 THEN 1730
1770 DISP USING 140 ; 'Dir ',COS(G),'x +
      ',SIN(G),'y = ',H1-C1
1780 W$=FNW$ @ IF NUM(W$)=8 THEN 1750
1790 GOTO 1950
1800 S=N^2-4*L*F
1810 IF S>=0 THEN 1850                       -No real solution
1820 !
1830 DISP 'No real solution set.' @ GOSU
     B 1990
1840 GOTO 1950
1850 IF S>0 THEN 1890
1860 !                                       -Straight line
1870 DISP USING 140 ; 'Line: ',-SIN(G),'
     x + ',COS(G),'y = ',-N/(2*L)
1880 GOSUB 1990 @ GOTO 1950
1890 !                                       -Two parallel lines
1900 DISP 'Two parallel lines ...' @ GOS
     UB 1990
1910 DISP USING 140 ; 'Line1 ',-SIN(G),'
     x + ',COS(G),'y = ',(-N+SQR(S))/(2*
     L)
1920 W$=FNW$ @ IF NUM(W$)=8 THEN 1900
1930 DISP USING 140 ; 'Line2 ',-SIN(G),'
     x + ',COS(G),'y = ',(-N-SQR(S))/(2*
     L)
1940 W$=FNW$ @ IF NUM(W$)=8 THEN 1910
1950 INPUT 'Run again?','Y'; Q$ @ Q$=UPR
     C$(Q$[1,1])
1960 IF Q$='Y' THEN 160
1970 DISP '          END OF PROGRAM'
1980 STOP
1990 !                                       -Wait for RTN key
2000 IF NUM(KEY$)#13 THEN 2000
2010 RETURN
```

PROGRAM DESCRIPTION

PARABOLIC EQUATIONS

This program determines the equation of the parabola passing through three points entered by the user. If the equation cannot be determined by this program or the points do not represent a function, the program will so indicate.

The program is designed to find equations of parabolas having a vertical axis, using the standard $y=Ax^2 + Bx + C$.

SAMPLE PROBLEM

Find an equation for the following parabolas represented by three points.

1) -4,3 0,1 4,3

2) -6,-4 -3,-1 0,-4

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		PARABOLIC EQUATIONS	
1	Enter 3 points on parabola	First point x,y =	-4,3 [RTN]
		Second point x,y =	0,1 [RTN]
		Third point x,y =	4,3 [RTN]
2	Display equation	y = .125x^2 + 0x + 1	[RTN]
3	Run again for problem #2	Run again, View again, or End? R	[RTN]
4	Enter 3 points	First point x,y =	-6,-4 [RTN]
		Second point x,y =	-3,-1 [RTN]
		Third point x,y =	0,-4 [RTN]
5	Display equation	y = -.333x^2 + -2x + -4	[RTN]
6	End	Run again, View again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		PARABOLIC EQUATIONS	
1	Enter coordinates of 3 points	First point x,y =	x_1,y_1 [RTN]
	on the parabola	Second point x,y =	x_2,y_2 [RTN]
		Third point x,y =	x_3,y_3 [RTN]
2	Display equation	y = n.nnnx^2 + .nnnx + n.nnn	[RTN]
3	Program options	Run again, View again, or End? R	V,E or [RTN]
	If 'R' then step 1		
	If 'V' then step 2		
	If 'E' then end	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A	Coefficient of x^2 in final equation	X1,Y1	Coordinates of points on parabola entered by user.
B	Coefficient of x in final equation	X2,Y2	
D	Preliminary calculation for coefficients also used in determining validity of data	X3,Y3	
		F1,F2,F3	Preliminary calculations for constant C

NOTES AND REFERENCES

Reference: PARABOLIC EQUATION, HP-85 MATH Solution Book Series 80, Hewlett Packard, 1980.

Note: Points on the parabola may be entered in any sequence.

PROGRAM LISTING

```
 10 ! PARABO - Finds equation
 20 ! of parabola passing
 30 ! through three points.
 40 ! Revision 11/01/82.
 50 !
 60 DIM Q$[4]
 70 REAL A,B,F1,F2,F3,C,D
 80 SHORT X1,X2,X3,Y1,Y2,Y3
 90 DEF FND(X)                                  -Function to round answers to
                                                 three places
100 IF X<0 THEN FND=-IP(-X*1000+.5)/100
    0 ELSE FND=IP(X*1000+.5)/1000
110 END DEF
120 DISP '      PARABOLIC EQUATIONS' @
    WAIT 2
130 ON ERROR GOTO 140
140 INPUT 'First point  x,y = ';X1,Y1
150 INPUT 'Second point  x,y = ';X2,Y2
160 INPUT 'Third point  x,y = ';X3,Y3
170 D=X2*X1^2+X1*X3^2+X3*X2^2-X3*X1^2-X         -Intermediate result - also for
    1*X2^2-X2*X3^2                              error trapping
180 IF D#0 THEN 230
190 IF X1#X2 OR X2#X3 THEN DISP 'Equati
    on cannot be determined.' @ GOTO 21
    0
200 DISP 'Nonfunction.  X ='X1                  -Display error message
210 IF NUM(KEY$)#13 THEN 210
220 GOTO 360
230 OFF ERROR
240 A=(Y1*(X2-X3)+Y2*(X3-X1)+Y3*(X1-X2)         -Compute A coefficient
    )/D
250 B=(Y1*(X3^2-X2^2)+Y2*(X1^2-X3^2)+Y3         -Compute B coefficient
    *(X2^2-X1^2))/D
260 F1=X3*X2^2-X2*X3^2                          -F1, F2, and F3 are
                                                 intermediate results for C
270 F2=X1*X3^2-X3*X1^2
280 F3=X2*X1^2-X1*X2^2
290 C=(Y1*F1+Y2*F2+Y3*F3)/D                     -Compute C
300 !
310 ! OUTPUT EQUATION.
320 !
330 DISP USING 340 ; FND(A),FND(B),FND(         -Display results
    C)
340 IMAGE 'y = ',K,'x^2 +',K,'x +',K
350 IF NUM(KEY$)#13 THEN 350
360 ON ERROR GOTO 370
370 DISP CHR$(210);'un again, ';CHR$(21         -Continuation options
    4);'iew again, or ';CHR$(197);
380 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
390 ON POS('RVE',Q$)+1 GOTO 360,130,310
    ,400
400 DISP '          END OF PROGRAM' @ WA
    IT 2 @ DISP
410 OFF ERROR
420 STOP
```

PROGRAM DESCRIPTION

ROOTS OF POLYNOMIALS

This program finds roots of polynomials using Barstow's method of iteration. The user provides the order (highest numbered exponent) of the polynomial and its coefficients in order from left to right.

Example: For $Ax^2 + Bx + C = 0$, enter the order, 2, then the coefficients A, B, and C as prompted. If one or more of the coefficients is zero, as in $9x^2 - 16$, enter 9, 0, -16 for the coefficients.

The roots of some forms of polynomials cannot be determined by this program and it will indicate so if true. Execution time for polynomials of high order may be excessive since many iterations may be required.

In cases where the program is not converging to a solution, the user will have the opportunity to end the computations for that polynomial or continue for 25 iterations.

SAMPLE PROBLEM

Determine the roots of the following polynomials:

1) $X^2 + X + 1$

2) $X^2 + 2X$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		ROOTS OF POLYNOMIALS	
1	Enter order (highest numbered	Order of polynomial?	2 [RTN]
	exponent)		
2	Enter coefficients	Coefficient #1 =	1 [RTN]
		Coefficient #2 =	1 [RTN]
		Coefficient #3 =	1 [RTN]
3	Display roots (roots are	Imag. root: -5 ± .87 * i	[RTN]
	complex for this polynomial)		
4	Run again for problem #2	Run again, or End? R	[RTN]
5	Enter order	Order of polynomial?	2 [RTN]
6	Enter coefficients	Coefficient #1 =	1 [RTN]
		Coefficient #2 =	2 [RTN]
		Coefficient #3 =	0 [RTN]
7	Display roots	Real root: 0	[RTN]
		Real root: -2	[RTN]
8	End	Run again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		ROOTS OF POLYNOMIALS	
1	Enter order of polynomial	Order of polynomial?	N [RTN]
2	Enter coefficients in order	Coefficient #1 =	n [RTN]
	from left to right	Coefficient #2 =	n [RTN]
		Coefficient #N+1	n [RTN]
3	Compute roots		
3a	If roots are too small for		
	computer to retain then	Solution unobtainable	[RTN]
	and goto step 5		
3b	If roots undetermined after		
	25 iterations then	No convergence; nn iterations	[RTN]
	option to continue computations	Continue for 25 iterations? Y	N or [RTN]
	If 'Y' then continue computa-		
	tions and goto step 3		
	If 'N' then goto step 5		
4	Display roots:		
	If roots are real then	Real root: n	[RTN]
	If roots are imaginary then	Imag. root: n±n*i	[RTN]
5	Program options	Run again, or End? R	E or [RTN]
	If 'R' then goto step 1		
	If 'E' then end	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
N	Order of polynomial	P1,Q1	Used to test convergence
A( )	Array for entering coefficients	I	Internal looping, index for subscripts
B( ),X( )	Temporary storage and manipulation of coefficients	J	Adjust subscript
		T,T1	Used to output solutions
D1	Number of sets of 25 iterations	X,X1,F,F1	Temporary storage for subroutine, used to test convergence in determining roots for odd exponents
C1	Number of iterations		
P,D	Solutions	Q$	User interaction
Q,R,S	Temporary storage		

NOTES AND REFERENCES

References: ROOTS OF POLYNOMIALS, HP-85 Numerical Analysis Solution Book, Hewlett Packard, 1980.

This program accepts polynomials with a maximum order of 10.

$$(Ax^{10} + Bx^{9} + Cx^{8} + Dx^{7} + Ex^{6} + Fx^{5} + Gx^{4} + Hx^{3} + Ix^{2} + Jx + K = \emptyset)$$

PROGRAM LISTING

```
 10 ! ROOTS - Finds roots of
 20 ! polynomials.
 30 !
 40 ! Revision 11/01/82.
 50 !
 60 DIM A(11),B(11),X(11)
 70 REAL P,P1,Q,Q1,F,F1,X,X1,T,T1,D
 80 INTEGER N,I,C1,J,D1
 90 DIM Q$[4]
100 DEF FND(X)                          -Function to round to two
                                         decimal places
110 IF X<0 THEN X=ABS(X) ELSE 130
120 X=IP(X*100+.5)/100 @ FND=-X @ GOTO
    140
130 FND=IP(X*100+.5)/100
140 END DEF
150 DISP '       ROOTS OF POLYNOMIALS'
160 D,Q,P=0
170 INPUT 'Order of polynomial?';N
180 IF N<=0 OR N>10 OR N#IP(N) THEN DIS
    P 'Invalid order' @ GOTO 170
190 FOR I=1 TO N+1
200 DISP 'Coefficient #';I;
210 INPUT '=';A(I)
220 B(I)=A(I)
230 NEXT I
240 C1=0
250 D1=1
260 IF N<=2 THEN 870
270 IF A(N+1)=0 THEN 950                -If rightmost coefficient is 0
                                         then one root is zero
280 IF N/2-IP(N/2)=0 THEN 310           -Test of N is even number
290 GOSUB 1100                          -Find root for odd numbered
                                         exponent
300 GOTO 260
310 IF ABS(A(N-1))<1.E-25 THEN 350
320 P=A(N)/A(N-1)
330 Q=A(N+1)/A(N-1)
340 GOTO 370
350 P=A(N)
360 Q=A(N+1)
370 FOR I=1 TO N+1
380 X(I)=A(I)
390 NEXT I
400 GOSUB 1040
410 FOR I=1 TO N-1
420 B(I)=X(I)
430 NEXT I
440 R=X(N)
450 S=A(N+1)-P*X(N)-Q*X(N-1)
460 GOSUB 1040
470 X(N)=P*X(N-1)-Q*X(N-2)
480 D=X(N-1)^2-X(N)*X(N-2)
490 IF ABS(D)>1.E-25 THEN 510
```

PROGRAM LISTING

```
500 DISP 'Solution unobtainable.' @ GOS
    UB 1440 @ GOTO 980
510 P1=P+(R*X(N-1)-S*X(N-2))/D
520 Q1=Q+(S*X(N-1)-R*X(N))/D
530 IF ABS(P)>1.E-25 THEN 570
540 IF ABS(P1)>1.E-25 THEN 570
550 IF ABS(Q)>1.E-25 THEN 580
560 GOTO 590
570 IF ABS(P1/P-1)>.000001 THEN 590          -Tests for convergence
580 IF ABS(Q1/Q-1)<.000001 THEN 680
590 P=P1
600 Q=Q1
610 C1=C1+1                                  -Increment iteration counter
620 IF C1=D1*25 THEN 640
630 GOTO 370
640 DISP 'No convergence;';C1;'iteratio
    ns.' @ GOSUB 1440
650 INPUT 'Continue for 25 iterations?'
    ,'Y'; Q$ @ Q$=UPRC$(Q$[1,1])
660 IF Q$='Y' THEN D1=D1+1 @ GOTO 370
670 GOTO 980
680 FOR I=2 TO N-1
690 A(I)=B(I)
700 NEXT I
710 N=N-2
720 D=P*P-4*Q
730 IF D<0 THEN 820
740 D=SQR(D)
750 ! DISPLAY ROOTS.
760 T=(-P+D)/2 @ GOSUB 1350                  -Display roots
770 T=(-P-D)/2 @ GOSUB 1350
780 C1=0
790 D1=1
800 IF N-2>0 THEN 270
810 GOTO 870
820 D=SQR(-D)
830 T=-P/2 @ T1=D/2 @ GOSUB 1390
840 C1=0
850 D1=1
860 IF N-2>0 THEN 270
870 IF N=1 THEN 930
880 IF N=0 THEN 980
890 P=B(2)/B(1)
900 Q=B(3)/B(1)
910 N=0
920 GOTO 720
930 T=-B(2)/B(1) @ GOSUB 1350
940 GOTO 980
950 T=0 @ GOSUB 1350
960 N=N-1
970 GOTO 260
980 !
990 DISP CHR$(210);'un again, or ';CHR$      -Continuation options
    (197);
```

PROGRAM LISTING

```
1000 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
     ,1])
1010 ON POS('RE',Q$)+1 GOTO 990,160,1020
1020 DISP '          END OF PROGRAM'
1030 STOP
1040 !
1050 X(2)=X(2)-P*X(1)
1060 FOR I=3 TO N
1070 X(I)=X(I)-P*X(I-1)-Q*X(I-2)
1080 NEXT I
1090 RETURN
1100 !
1110 IF B(2)=0 THEN 1140                    -Find root for odd numbered
                                             exponent
1120 X=-B(2)/B(1)
1130 GOTO 1150
1140 X=-B(N+1)/B(1)
1150 F=0
1160 F1=0
1170 FOR I=1 TO N+1
1180 J=N-I+2
1190 IF B(J)=0 THEN 1230
1200 F=B(J)*X^(I-1)+F
1210 IF I-1=0 THEN 1230
1220 F1=(I-1)*B(J)*X^(I-2)+F1
1230 NEXT I
1240 X1=X-F/F1
1250 IF ABS(X/X1-1)<.000001 THEN 1280       -Test for convergence
1260 X=X1
1270 GOTO 1150
1280 T=X1 @ GOSUB 1350
1290 N=N-1
1300 FOR I=2 TO N+1
1310 A(I)=B(I)+X1*A(I-1)
1320 B(I)=A(I)
1330 NEXT I
1340 RETURN
1350 !                                      -Display single real root
1360 DISP 'Real root: ';FND(T)
1370 GOSUB 1440
1380 RETURN
1390 !
1400 DISP 'Imag. root: '; @ DISP USING 1    -Display complex root
     420 ; FND(T),CHR$(171),FND(T1)
1410 GOSUB 1440
1420 IMAGE K,' ',K,' ',K,' * i'
1430 RETURN
1440 !
1450 IF NUM(KEY$)<>13 THEN 1450             -Wait for return key
1460 RETURN
```

PROGRAM DESCRIPTION

TRIANGLE SOLUTIONS

This program finds dimensions for unknown sides and angles of a triangle and calculates the area of the triangle. The user may select either degree or radian mode. When using degree mode, angles must be entered in decimal format (i.e., 98°12' is entered as 98.2). Results will also be in decimal format.

The user must enter three parts of the triangle corresponding to one of the five common triangle solutions described below:

EQUATIONS: (See Diagram A)

$S_1S_2S_3$ (all sides known) $A_3 = 2\ COS^{-1}\sqrt{\frac{P(P-S_2)}{S_1S_3}}$ $P = (S_1 + S_2 + S_3)/2$

$$A_2 = 2\ COS^{-1}\sqrt{\frac{P(P-S_1)}{S_2S_3}}$$

$$A_1 = COS^{-1}(-COS(A_3 + A_2))$$

$A_1S_1A_3$ (2 angles and enclosed side known) $A_2 = COS^{-1}(-COS(A_3 + A_1))$

$$S_2 = S_1\frac{Sin\ A_3}{Sin\ A_2}$$

$$S_3 = S_1COS\ A_3 + S_2COS\ A_2$$

$S_1A_1A_2$ (side and following 2 angles known) $A_3 = COS^{-1}(-COS(A_1 + A_2))$

Problem has been reduced to ASA configuration.

$S_1A_1S_2$ (2 sides and enclosed angle known) $S_3 = \sqrt{S_1^2 + S_2^2 - 2S_1S_2COS\ A_1}$

Problem has been reduced to SSS configuration.

PROGRAM DESCRIPTION

TRIANGLE SOLUTIONS (continued)

$S_1S_2A_2$ (2 sides and adjacent angle known)* $A_3 = SIN^{-1}\left[\frac{S_2}{S_1} SIN\ A_2\right]$

$A_1 = COS^{-1}\left[-COS(A_2 + A_3)\right]$

Problem has been reduced to ASA configuration.

* Note that two possible solutions exist if S_2 is greater than S, and $A_3 \neq 90°$. Both possible answer sets are calculated. (See Diagram B).

Diagram A

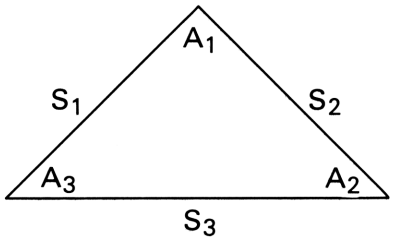
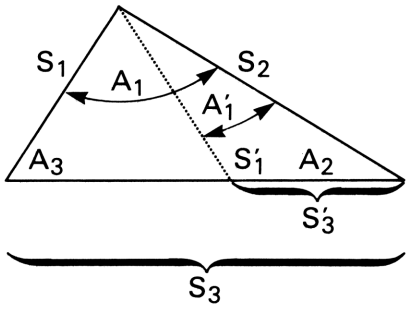


Diagram B



Area $= \frac{1}{2} S_1 S_3 \sin A_3$

SAMPLE PROBLEM

Solve the following triangles:

1) Side 1 = 3 Side 2 = 4 Side 3 = 5 (use SSS)

2) Angle 1 = 45° Side 1 = 10 Angle 3 = 10° (use ASA)

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		TRIANGLE SOLUTION	
1	Choose mode	Degree or Radian mode?	D [RTN]
		OPTIONS	
2	Choose SSS	1)SSS 2)ASA 3)SAA 4)SAS 5)SSA	1 [RTN]
3	Enter sides	Side 1 =	3 [RTN]
	Use [BACK] to display the	Side 2 =	4 [RTN]
	previous result	Side 3 =	5 [RTN]
4	Results	Side 1 = 3.00	[RTN]
		Angle 1 = 90.00	[RTN]/[BACK]
		Side 2 = 4.00	[RTN]/[BACK]
		Angle 2 = 36.87	[RTN]/[BACK]
		Side 3 = 5.00	[RTN]/[BACK]
		Angle 3 = 53.13	[RTN]/[BACK]
		Area = 6.00	[RTN]/[BACK]
5	Run again for problem #2	Run again, View again, or End? R	[RTN]
6	Choose mode	Degree or Radian mode?	D [RTN]
		OPTIONS	
7	Choose ASA	1)SSS 2)ASA 3)SAA 4)SAS 5)SSA	2 [RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
8	Enter parameters	Angle 1 =	45 [RTN]
		Side 1 =	10 [RTN]
		Angle 3 =	10 [RTN]
9	Results	Side 1 = 10.00	[RTN]
		Angle 1 = 45.00	[RTN]/[BACK]
		Side 2 = 2.12	[RTN]/[BACK]
		Angle 2 = 125.00	[RTN]/[BACK]
		Side 3 = 8.63	[RTN]/[BACK]
		Angle 3 = 10.00	[RTN]/[BACK]
		Area = 7.49	[RTN]/[BACK]
10	End	Run again, View again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		TRIANGLE SOLUTION	
1	Choose mode	Degree or Radian mode?	D or R [RTN]
2	Choose triangle solution	OPTIONS	
		1)SSS 2)ASA 3)SAA 4)SAS 5)SSA	1,2,3,4 or 5 [RTN]
	If '1' (SSS) then step 3		
	If '2' (ASA) then step 4		
	If '3' (SAA) then step 5		
	If '4' (SAS) then step 6		
	If '5' (SSA) then step 7		
3	SSS-enter sides	Side 1 =	S_1 [RTN]
	Solve for missing angles	Side 2 =	S_2 [RTN]
	and goto step 8	Side 3 =	S_3 [RTN]
4	ASA-enter two angles and	Angle 1 =	A_1 [RTN]
	enclosed side. Solve for	Side 1 =	S_1 [RTN]
	missing angle and sides and	Angle 3 =	A_3 [RTN]
	goto step 8		
5	SAA-enter side and two	Side 1 =	S_1 [RTN]
	following angles. Solve for	Angle 1 =	A_1 [RTN]
	missing angle and sides and	Angle 2 =	A_2 [RTN]
	goto step 8		
6	SAS-enter two sides and	Side 1 =	S_1 [RTN]
	enclosed angle. Solve for	Angle 1 =	A_1 [RTN]
	missing side and angles and	Side 2 =	S_2 [RTN]
	goto step 8		
7	SSA-enter two sides	Side 1 =	S_1 [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	and the adjacent angle. Solve	Side 2 =	S_2 [RTN]
	for missing side and angles.	Angle 2 =	A_2 [RTN]
8	Results:	Side 1 = sss.ss	[RTN]
	Use [BACK] to display the	Angle 1 = aa.aa	[RTN]/[BACK]
	previous result	Side 2 = sss.ss	[RTN]/[BACK]
		Angle 2 = aaa.aa	[RTN]/[BACK]
		Side 3 = sss.ss	[RTN]/[BACK]
		Angle 3 = aa.aa	[RTN]/[BACK]
		Area = nnn.nn	[RTN]/[BACK]
8a	If more than one solution		
	then goto step 8 and view		
	second answer set		
9	Program options	Run again, View again, or End? R	V, E or [RTN]
	If 'R' then step 1		
	If 'V' then step 8		
	If 'E' then End	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
F,F1	Flags to indicate type of triangle solution chosen	H	Area of triangle
S1,S2,S3	Lengths of sides 1,2, and 3 of triangle	M$	Degree or Radian mode
A1,A2,A3	Angles 1,2, and 3 (in decimal degrees)	N$	Choice of triangle solutions
P	Half the perimeter	Q$	User interaction

NOTES AND REFERENCES

References: TRIANGLE SOLUTIONS, HP-85 MATH PACK, Hewlett Packard, 1980.

The accuracy of solutions may degenerate for triangles having extremely small angles.

PROGRAM LISTING

```
 10 ! TRIANG - Solves triangles
 20 ! in degree or radian mode.
 30 !
 40 ! Revision 11/01/82.
 50 !
 60 INTEGER F,F1
 70 DIM M$[6],N$[3],Q$[2]
 80 REAL A1,A2,A3,S1,S2,S3,P,H
 90 DEF FNQ$                                  -Wait for RTN or BACK key
100 Z=NUM(KEY$) @ IF Z#13 AND Z#8 THEN
    100
110 FNQ$=CHR$(Z)
120 END DEF
130 DISP '        TRIANGLE SOLUTION' @ W
    AIT 2
140 DISP CHR$(196);'egree or ';CHR$(210   -Select degree or radian mode
    );
150 ON ERROR GOTO 140
160 INPUT 'adian mode?'; M$ @ M$=UPRC$(
    M$[1,1])
170 OFF ERROR
180 IF M$='D' THEN OPTION ANGLE DEGREES
190 IF M$='R' THEN OPTION ANGLE RADIANS
200 F1=0
210 DISP TAB(13);'OPTIONS' @ WAIT 1
220 ON ERROR GOTO 230
230 INPUT '1)SSS 2)ASA 3)SAA 4)SAS 5)SS   -Select problem to solve
    A ';N$
240 N$=UPRC$(N$[1,1])
250 OFF ERROR
260 ON POS('12345',N$)+1 GOTO 230,270,4
    00,500,570,650
270 ! SSS.                                  -Side-side-side
280 ON ERROR GOSUB 1100 @ GOTO 290
290 INPUT 'Side 1 = ';S1
300 INPUT 'Side 2 = ';S2
310 INPUT 'Side 3 = ';S3
320 F=0
330 P=(S1+S2+S3)/2                          -Compute half the perimeter.
340 A3=2*ACOS(SQR(P*(P-S2)/(S1*S3)))
350 A2=2*ACOS(SQR(P*(P-S1)/(S2*S3)))
360 IF F=1 THEN 380
370 GOSUB 880
380 OFF ERROR @ GOSUB 920
390 GOTO 810
400 ! ASA.                                  -Angle-side-angle
410 ON ERROR GOSUB 1100 @ GOTO 420
420 INPUT 'Angle 1 = ';A1
430 INPUT 'Side 1 = ';S1
440 INPUT 'Angle 3 = ';A3
450 A2=ACOS(-COS(A3+A1))
460 S2=S1*SIN(A3)/SIN(A2)
470 S3=S1*COS(A3)+S2*COS(A2)
480 OFF ERROR @ GOSUB 920
```

PROGRAM LISTING

```
490 GOTO 810
500 ! SAA.                                    -Side-angle-angle
510 ON ERROR GOSUB 1100 @ GOTO 520
520 INPUT 'Side 1 = ';S1
530 INPUT 'Angle 1 = ';A1
540 INPUT 'Angle 2 = ';A2
550 A3=ACOS(-COS(A2+A1))
560 GOTO 460
570 ! SAS.                                    -Side-angle-side
580 ON ERROR GOSUB 1100 @ GOTO 590
590 INPUT 'Side 1 = ';S1
600 INPUT 'Angle 1 = ';A1
610 INPUT 'Side 2 = ';S2
620 S3=SQR(S1^2+S2^2-2*S1*S2*COS(A1))
630 F=1
640 GOTO 330
650 ! SSA.                                    -Side-side-angle (can yield two
                                               results)
660 ON ERROR GOSUB 1100 @ GOTO 670
670 INPUT 'Side 1 = ';S1
680 INPUT 'Side 2 = ';S2
690 INPUT 'Angle 2 = ';A2
700 A3=ASIN(S2/S1*SIN(A2))
710 GOSUB 880
720 S3=S1*COS(A3)+S2*COS(A2)
730 F1=0
740 OFF ERROR @ GOSUB 920
750 IF S2<=S1 THEN 810
760 F1=1
770 A3=ACOS(-COS(A3))
780 GOSUB 880
790 S3=S1*COS(A3)+S2*COS(A2)
800 GOSUB 920
810 DISP CHR$(210);'un again, ';CHR$(21    -Continuation options
    4);'iew again, or ';CHR$(197);
820 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
830 ON POS('RVE',Q$)+1 GOTO 810,140,840
    ,850
840 IF F1=1 THEN 700 ELSE GOSUB 920 @ G
    OTO 810
850 DISP '          END OF PROGRAM'
860 DISP
870 STOP
880 !                                         -Find third angle given first
                                               two
890 A1=ACOS(-COS(A3+A2))
900 F=0
910 RETURN
920 ! OUTPUT.                                 -Display results
930 H=S1*S3*SIN(A3)/2
940 DISP USING 950 ; 'Side  1 =',S1
950 IMAGE 10a,dddd.dd
960 IF NUM(KEY$)#13 THEN 960
```

PROGRAM LISTING

```
 970 DISP USING 950 ; 'Angle 1 =',A1
 980 Q$=FNQ$ @ IF NUM(Q$)=8 THEN 940
 990 DISP USING 950 ; 'Side  2 =',S2
1000 Q$=FNQ$ @ IF NUM(Q$)=8 THEN 970
1010 DISP USING 950 ; 'Angle 2 =',A2
1020 Q$=FNQ$ @ IF NUM(Q$)=8 THEN 990
1030 DISP USING 950 ; 'Side  3 =',S3
1040 Q$=FNQ$ @ IF NUM(Q$)=8 THEN 1010
1050 DISP USING 950 ; 'Angle 3 =',A3
1060 Q$=FNQ$ @ IF NUM(Q$)=8 THEN 1030
1070 DISP USING 950 ; 'Area =',H
1080 Q$=FNQ$ @ IF NUM(Q$)=8 THEN 1050
1090 RETURN
1100 !                                    -Display error message
1110 DISP 'Not a triangle.  Re-enter dat
     a.'
1120 WAIT 2
1130 RETURN
```

PROGRAM DESCRIPTION

POLYGON AREA

This program finds the area enclosed in any polygon given the coordinates of its vertices. First the user must enter the number of vertices of the polygon, and then enter the X,Y coordinates of each of these vertices. The coordinates must be entered in sequential order, either clockwise or counterclockwise. The area is then displayed and the user has the opportunity to run the program again.

SAMPLE PROBLEM

Find the area of the polygon having vertices at

Ø,3 4,4 6,2 4,Ø 5,-3 Ø,-2

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		POLYGON AREA	
1	Enter number of vertices	Number of points =	6 [RTN]
2	Enter coordinates of vertices	X,Y #1 =	0,3 [RTN]
	in sequential order	X,Y #2 =	4,4 [RTN]
		X,Y #3 =	6,2 [RTN]
		X,Y #4 =	4,0 [RTN]
		X,Y #5 =	5,-3 [RTN]
		X,Y #6 =	0,-2 [RTN]
3	Output	The area is = 29	[RTN]
4	End	Run again, View again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		POLYGON AREA	
1	Enter number of vertices	Number of points =	n [RTN]
2	Enter vertices X,Y in	X,Y #1 =	X_1,Y_1 [RTN]
	sequential order	X,Y #2 =	X_2,Y_2 [RTN]
		X,Y #n =	Xn,Yn [RTN]
3	Compute area and		
3a	display result	The area is nn.nn	[RTN]
4	Program options	Run again, View again, or End? R	V,E or [RTN]
4a	If 'R' then step 1		
4b	If 'V' then step 3a		
4c	If 'E' then end program	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A1	Total polygon area	Z	Number of points to be entered
D1,D2	Difference between initial point and successive points	Z1	Counts number of points
		Q$	User interaction
D3,D4	Used in computing triangle areas (adjusts polygon to X-Y grid)	A	Finds area of triangular section of polygon

NOTES AND REFERENCES

References: POLYGON AREA, HP-85 Math Solutions Book, Series 80, Hewlett Packard, 1980.

PROGRAM LISTING

```
 10 ! POLYGN - Finds area
 20 ! of a polygon.
 30 !
 40 ! Revision 11/01/82.
 50 !
 60 REAL A1,D1,D2,D3,D4,A
 70 INTEGER Z1,Z
 80 DIM Q$[3]
 90 DEF FND(X) = IP(X*100+.5)/100          -Function to round answers to
                                            two decimal places
100 DISP '           POLYGON AREA' @ WAI
    T 2
110 INPUT 'Number of points = ';Z
120 IF Z<3 THEN 110
130 A1=0
140 !
150 ! ENTER VERTICES OF POLYGON.
160 !
170 INPUT 'X,Y  # 1 =';X(1),Y(1)
180 INPUT 'X,Y  # 2 =';X(2),Y(2)
190 D1=X(2)-X(1)
200 D2=Y(2)-Y(1)
210 Z1=3
220 DISP 'X,Y  #';Z1; @ INPUT ' =';X(Z1
    ),Y(Z1)
230 D3=X(Z1)-X(1)
240 D4=Y(Z1)-Y(1)
250 A=(D2*D3-D1*D4)/2                      -Finds the area of triangular
                                            section of polygon
260 A1=A1+A                                -Sums triangle areas to find
                                            total area
270 D1=D3
280 D2=D4
290 IF Z1=Z THEN 330
300 Z1=Z1+1
310 GOTO 220
320 !
330 ! OUTPUT AREA.
340 !
350 DISP 'The area is ';FND(ABS(A1))
360 IF NUM(KEY$)#13 THEN 360               -Wait for RTN key to continue
370 DISP CHR$(210);'un again, ';CHR$(21    -Continuation options
    4);'iew again, or ';CHR$(197);
380 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
390 ON POS('RVE',Q$)+1 GOTO 370,110,330
    ,400
400 DISP '          END OF PROGRAM'
410 STOP
```

PROGRAM DESCRIPTION

HYPERBOLIC FUNCTIONS

This program solves hyperbolic sine, cosine, tangent and their inverses. The user must choose the desire function by:

S = hyperbolic sine	AS = inverse hyperbolic sine
C = hyperbolic cosine	AC = inverse hyperbolic cosine
T = hyperbolic tangent	AT = inverse hyperbolic tangent

and enter the parameter for the function.

The program uses the following equations:

HYPERBOLIC FUNCTIONS

$$\sinh(x) = \frac{(e^{x} - e^{-x})}{2}$$

$$\cosh(x) = \frac{(e^{x} + e^{-x})}{2}$$

$$\tanh(x) = 1 - \frac{2}{1 + e^{2x}}$$

INVERSE HYPERBOLIC FUNCTIONS

$$\sinh^{-1}(x) = \mathrm{Ln}[x + (x^{2}+1)^{\frac{1}{2}}]$$

$$\cosh^{-1}(x) = \mathrm{Ln}[x + (x^{2}-1)^{\frac{1}{2}}],\ x \geq 1$$

$$\tanh^{-1}(x) = \tfrac{1}{2}\,\mathrm{Ln}[\frac{1+x}{1-x}],\ x^{2}<1$$

SAMPLE PROBLEM

Determine these hyperbolic functions:

1) SINH (3.5)
2) ACOSH (45)

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		HYPERBOLIC FUNCTIONS	
1	Choose hyperbolic sine	Function (A)SCT?	S [RTN]
2	Enter X parameter	X=	3.5 [RTN]
3	Display result	SIHN(3.5) = 16.5426272877	[RTN]
4	Run again for problem #2	Run again, View again, or End? R	[RTN]
5	Choose inverse hyperbolic	Function (A)SCT?	AC [RTN]
	cosine		
6	Enter X parameter	X =	45 [RTN]
7	Display result	ACOSH(45) = 4.49968619067	[RTN]
8	End	Run again, View again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
		HYPERBOLIC FUNCTIONS	
1	Choose function	Function (A)SCT?	S, C, T,
	S - SINH		AS, AC, or
	C - COSH		AT [RTN]
	T - TANH		
	AS - ASINH		
	AC - ACOSH		
	AT - ATANH		
2	Enter parameter (X)	X =	X [RTN]
2a	On function 'S' goto step 3		
2b	On function 'C' goto step 4		
2c	On function 'T' goto step 5		
2d	On function 'AS' goto step 6		
2e	On function 'AC' goto step 7		
2f	On function 'AT' goto step 8		
3	Hyperbolic Sine	SINH(X) = n.nnnnnnnnnn	[RTN]
	Goto step 9		
4	Hyperbolic Cosine	COSH(X) = n.nnnnnnnnnn	[RTN]
	Goto step 9		
5	Hyperbolic Tangent	TANH(X) = n.nnnnnnnnnn	[RTN]
	Goto step 9		
6	Inverse hyperbolic Sine	ASINH(X) = n.nnnnnnnnnn	[RTN]
	Goto step 9		
7	Inverse hyperbolic Cosine	ACOSH(X) = n.nnnnnnnnnn	[RTN]
	Goto step 9		

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
8	Inverse hyperbolic Tangent	ATANH(X) = n.nnnnnnnnnn	[RTN]
9	Program options	Run again, View again, or End? R	V,E or [RTN]
	If 'R' then step 1		
	If 'V' then step 2a		
	If 'E' then end	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
X	Parameter for function	F$	Choice of functions
H	Result of function on parameter	F1$	Label output with chosen function
F	Flag for hyperbolic SINE to signal parameter less than zero.		

NOTES AND REFERENCES

References: HYPERBOLICS, HP-85 Math Pac, Hewlett Packard, 1980.

PROGRAM LISTING

```
 10 ! HYPER - Computes
 20 ! hyperbolic functions.
 30 ! Revision 11/01/82.
 40 DIM F$[5],F1$[5],Q$[4]
 50 REAL X,H
 60 DISP '     HYPERBOLIC FUNCTIONS'
 70 ON ERROR GOTO 80
 80 INPUT 'Function (I)SCT?',' '; F$ @
    F$=UPRC$(F$)                          -Select function
 90 IF LEN(F$)=1 THEN F$=' '&F$
100 ON POS(' S C TISICIT',F$)+1 GOTO 80
    ,160,80,280,80,330,80,380,80,430,80
    ,490,80
110 ! ENTER PARAMETER.                    -Enter parameter
120 ON ERROR GOTO 130
130 INPUT 'X =';X
140 OFF ERROR
150 RETURN
160 ! SINH
170 GOSUB 110
180 IF X<0 THEN F=1 ELSE F=0
190 IF X=0 THEN H=0 @ GOTO 260
200 X=ABS(X)
210 IF X>.5 THEN 240
220 H=X*(EXP(X)-1)/LOG(EXP(X))*(EXP(-X)   -H contains the result for each
    +1)/2                                  function of X
230 GOTO 250
240 H=(EXP(X)-EXP(-X))/2
250 IF F=1 THEN H=-H @ X=-X
260 F1$='SINH'                            -F1$ contains the name of the
                                           function
270 GOTO 540
280 ! COSH.
290 GOSUB 110
300 H=(EXP(X)+1/EXP(X))/2
310 F1$='COSH'
320 GOTO 540
330 ! TANH.
340 GOSUB 110
350 H=1-2/(1+EXP(2*X))
360 F1$='TANH'
370 GOTO 540
380 ! ASINH.
390 GOSUB 110
400 H=LOG(X+SQR(X*X+1))
410 F1$='ASINH'
420 GOTO 540
430 ! ACOSH.
440 GOSUB 110
450 IF X<1 THEN GOSUB 640 @ GOTO 580
460 H=LOG(X+SQR(X*X-1))
470 F1$='ACOSH'
480 GOTO 540
490 ! ATANH.
```

PROGRAM LISTING

```
500 GOSUB 110
510 IF X^2=1 THEN GOSUB 640 @ GOTO 580
520 H=LOG((1+X)/(1-X))/2
530 F1$='ATANH'
540 ! OUTPUT.
550 DISP USING 560 ; F1$,X,H                    -Display results
560 IMAGE k,'(',k,')=',K
570 IF NUM(KEY$)#13 THEN 570                    -Wait for RTN key
580 ON ERROR GOTO 590
590 DISP CHR$(210);'un again, ';CHR$(21         -Continuation options
    4);'iew again, or ';CHR$(197);
600 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
610 ON POS('RVE',Q$)+1 GOTO 580,70,540,
    620
620 OFF ERROR
630 DISP '          END OF PROGRAM' @ ST
    OP
640 DISP 'Invalid parameter.' @ WAIT 2          -Display error message
    @ RETURN
```

PROGRAM DESCRIPTION

COMPLEX TRIGONOMETRIC FUNCTIONS

This program will compute the sine, cosine, and tangent of a complex number of the form X + Yi. The program will also compute the hyperbolic trigonometric functions. The function is selected from the following table:

S - sine	HS - hyperbolic sine
C - cosine	HC - hyperbolic cosine
T - tangent	HT - hyperbolic tangent

The formulae used are as follows:

$$\sin(z) = \sin(x)\cosh(y) + \cos(x)\sinh(y)i$$

$$\cos(z) = \cos(x)\cosh(y) - \sin(x)\sinh(y)i$$

$$\tan(z) = \frac{\sin(zx) + \sinh(zy)i}{\cos(zx) + \cosh(zy)}$$

$$\sinh(z) = \sinh(x)\cos(y) + \cosh(x)\sin(y)i$$

$$\cosh(z) = \cosh(x)\cos(y) + \sinh(x)\sin(y)i$$

$$\tanh(z) = \frac{\sinh(zx) + \sin(zy)i}{\cosh(zx) + \cos(zy)}$$

SAMPLE PROBLEM

1) Compute the sine of 3-i

2) Compute the hyperbolic cosine of -2 + 12i

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	Complex Trigonometric Functions	
2	Enter coefficients	Enter real coefficient?	3 [RTN]
		Enter imaginary coefficient?	-1 [RTN]
3	Select sine function	Function (H) SCT :	S [RTN]
		SINE	
4	Display result	.2178 + 1.1634i	[RTN]
5	Run again for problem 2	Run again, or End? R	[RTN]
6	Enter coefficents	Enter real coefficient?	-2 [RTN]
		Enter imaginary coefficient?	12 [RTN]
7	Select hyperbolic cosine	Function (H) SCT :	HC [RTN]
		HYPERBOLIC COSINE	
8	Display result	3.1747 + 1.9461i	[RTN]
9	End	Run again, or End? R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	Complex Trigonometric Functions	
2	Enter complex argument	Enter real coefficient?	a [RTN]
	a - real coefficient	Enter imaginary coefficient?	b [RTN]
	b - imaginary coefficient		
3	Select function to perform	Function (H)SCT=	S,
	'S' - sine (goto 4)		C,
	'C' - cosine (goto 5)		T,
	'T' - tangent (goto 6)		HS,
	'HS' - hyperbolic sine (goto 7)		HC,
	'HC' - hyperbolic cosine (goto 8)		or HT
	'HT' - hyperbolic tangent (goto 9)		[RTN]
4	Sine - compute and display re-	SINE	
	sult in complex form. Goto 10	n.nnnn + nn.nnnni	[RTN]
5	Cosine - compute and display.	COSINE	
	Goto 10	n.nnnn + nn.nnnni	[RTN]
6	Tangent - compute and display.	TANGENT	
	Goto 10	nn.nnnn + n.nnnni	[RTN]
7	Hyperbolic sine - compute and	HYPERBOLIC SINE	
	display. Goto 10	.nnnn + nn.nnnni	[RTN]
8	Hyperbolic cosine - compute	HYPERBOLIC COSINE	
	and display. Goto 10	n.nnnn + nn.nnnni	[RTN]
9	Hyperbolic tangent - compute	HYPERBOLIC TANGENT	
	and display. Goto 10	nn.nnnn + n.nnnni	[RTN]
10	Option to run again: Enter	Run again, or End? R	E or [RTN]
	'R' to run again, 'E' to end.	END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A	Real part of result	E	Denominator for hyperbolic tangent
B	Imaginary part of result	X	Real part of argument
D	Denominator for tangent	Y	Imaginary part of argument
F$	Choice of function	Q$	End of program options

NOTES AND REFERENCES

References: COMPLEX TRIGONOMETRIC FUNCTIONS, HP-85 Math, Series 80, Hewlett Packard, 1981.

Abramowitz, M. and Stegun, J. A., HANDBOOK OF MATHEMATICAL FUNCTIONS, National Bureau of Standards, 1965, p. 74, 84.

PROGRAM LISTING

```
 10 ! COMPLEX - Trigonometric
 20 ! functions of a
 30 ! complex argument.
 40 !
 50 ! Revision 11/01/82.
 60 DELAY 1
 70 DIM M$[4],F$[3],Q$[3]
 80 REAL A,B,X,Y,D,E
 90 DEF FNS(W) = (EXP(W)-EXP(-W))/2          -Hyperbolic sine used to
                                              compute other functions.
100 DEF FNC(W) = (EXP(W)+EXP(-W))/2          -Hyperbolic cosine used to
                                              compute other functions

110 DEF FND(W)
120 IF W<0 THEN FND=-IP(-W*10000+.5)/10
    000 ELSE FND=IP(W*10000+.5)/10000
130 END DEF
140 DISP ' Complex Trigonometric Functi
    ons'
150 OPTION ANGLE RADIANS
160 ON ERROR GOTO 170                        -Data may be entered in degree
                                              or radian mode

170 INPUT 'Enter real coefficient?';X        -Enter complex number
180 ON ERROR GOTO 190
190 INPUT 'Enter imaginary coefficient?
    ';Y
200 ON ERROR GOTO 210
210 INPUT 'Function (H)SCT:'; F$ @ F$=U      -Select function
    PRC$(F$)
220 IF LEN(F$)=1 THEN F$=' '&F$ ELSE F$
    =F$[1,2]
230 ON POS(' S C THSHCHT',F$)+1 GOTO 21
    0,240,210,290,210,340,210,400,210,4
    50,210,500,210
240 ! SINE.
250 DISP TAB(14);'SINE'
260 A=SIN(X)*FNC(Y)
270 B=COS(X)*FNS(Y)
280 GOTO 550
290 ! COSINE.
300 DISP TAB(13);'COSINE'
310 A=COS(X)*FNC(Y)
320 B=-SIN(X)*FNS(Y)
330 GOTO 550
340 ! TANGENT.
350 DISP TAB(12);'TANGENT'
360 D=COS(2*X)+FNC(2*Y)                      -Denomination for computing
                                              hyperbolic tangent

370 A=SIN(2*X)/D
380 B=FNS(2*Y)/D
390 GOTO 550
400 ! HYPERBOLIC SINE.
410 DISP TAB(8);'HYPERBOLIC SINE'
420 A=FNS(X)*COS(Y)
430 B=FNC(X)*SIN(Y)
```

PROGRAM LISTING

```
440 GOTO 550
450 ! HYPERBOLIC COSINE.
460 DISP TAB(7);'HYPERBOLIC COSINE'
470 A=FNC(X)*COS(Y)
480 B=FNS(X)*SIN(Y)
490 GOTO 550
500 ! HYPERBOLIC TANGENT.
510 DISP TAB(7);'HYPERBOLIC TANGENT'
520 E=FNC(2*X)+COS(2*Y)
530 A=FNS(2*X)/E
540 B=SIN(2*Y)/E
550 DISP USING 560 ; FND(A),FND(B)          -Display result
560 IMAGE K,' + ',K,'i'
570 IF NUM(KEY$)#13 THEN 570
580 ON ERROR GOTO 590
590 DISP CHR$(210);'un again, or ';CHR$     -Continuation options
    (197);
600 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
610 ON POS('RE',Q$)+1 GOTO 590,150,620
620 OFF ERROR
630 DISP TAB(9);'END OF PROGRAM'
640 STOP
```

PROGRAM DESCRIPTION

PRIME FACTORIZATION

This program tries as factors for the number N all numbers in the set:

[2, 3, 5, P where P $\equiv$ 1, 7, 11, 13, 17, 19, 23, 29 mod 30 and P$\leq$sqr(n)]

up to the square root of N. If a factor P is found, P^n is factored out of N and $N=N/P^n$ proceeding from P through the set of remaining test factors until a test factor exceeds SQR(N).

SAMPLE PROBLEM

Factor the numbers 37559, 3212453, and 976142.

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run Program	** PRIME FACTORIZATION **	
2	Enter number to be factored	Enter N?	37559 [RTN]
3	View factors	No. Factor Power	
		1 23 2	[RTN]
		2 71 1	[RTN]
4	Run again	Run again, View again, or End?R	[RTN]
5	Enter number to be factored	Enter N?	3212453 [RTN]
6	View factors	No. Factor Power	
	(Number is prime)	1 3212453 1	[RTN]
7	Run again	Run again, View again, or End?R	[RTN]
8	Enter number to be factored	Enter N?	9736142 [RTN]
	View factors	No. Factor Power	
		1 2 1	[RTN]
		2 13 1	[RTN]
		3 439 1	[RTN]
		4 853 1	[RTN]
10	End program.	Run again, View again, or End?R	E [RTN]
		END OF PROGRAM	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	** PRIME FACTORIZATION **	
2	Enter number to be factored	Enter N?	N [RTN]
3	View factors	No. Factor Power	
	[RTN] will advance factors,	# *factor* *power*	[RTN or BACK]
	[BACK] will show previous		
	factor.		
5	Continuation options	Run again, View again, or End?R	
5a	[RTN] will re-run program.		[RTN]
5b	V [RTN] will re-view factors		V [RTN]
5c	E [RTN] will end program.		E [RTN]
		END OF PROGRAM	

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
F	Number of factors	K	Current test factor
K()	Array of factors	P()	Array of powers
N	Number to be factored	S	Square root of N
I	Index for display loop	A,Q$	Control for options

NOTES AND REFERENCES

Notes: Numbers in excess of 10^{12} may not factor correctly.

Number with more than 10 distinct factors will overflow the factor array. To change this, modify the dimension statement in line 60. Note that the smallest number with 11 factors is 200560490130. This program factors the positive integral part of N.

Reference: Knuth, Donald E., SEMINUMERICAL ALGORITHMS, Addison-Wesley, 1980, Section 4.5.4.

PROGRAM LISTING

```
 10 ! Factor - Determines the
 20 ! prime factorization of a
 30 ! number.
 40 !
 50 ! Revision 11/01/82
 60 DIM K(10),P(10)
 70 DELAY 1
 80 DEF FNA
 90 A=NUM(KEY$) @ IF A#8 AND A#13 THEN
    90                                    -Function returns 8 for BACK
100 FNA=A @ END DEF                        key or 13 for RTN key
110 DISP '   * * PRIME FACTORIZATION *
    *'
120 INPUT 'Enter N?';N
130 N=ABS(INT(N))                         -Trap numbers that cannot be
                                           factored
140 IF N=0 OR N=1 THEN BEEP @ DISP 'Can
    not factor';N @ GOTO 120
150 F=0                                   -Initialize counter for number
                                           of factors
160 K=2 @ IF MOD(N,K)=0 THEN GOSUB 290    -Test first three factors
170 K=3 @ IF MOD(N,K)=0 THEN GOSUB 290
180 K=5 @ IF MOD(N,K)=0 THEN GOSUB 290
190 S=SQR(N)
200 K=K+2 @ IF K>S THEN 330               -Build successive factors to
                                           test. Stop if > SQR(N)
210 IF MOD(N,K)=0 THEN GOSUB 290
220 K=K+4 @ IF MOD(N,K)=0 THEN GOSUB 29
    0
230 K=K+2 @ IF MOD(N,K)=0 THEN GOSUB 29
    0
240 K=K+4 @ IF MOD(N,K)=0 THEN GOSUB 29
    0
250 K=K+2 @ IF MOD(N,K)=0 THEN GOSUB 29
    0
260 K=K+4 @ IF MOD(N,K)=0 THEN GOSUB 29
    0
270 K=K+6 @ IF MOD(N,K)=0 THEN GOSUB 29
    0
280 GOTO 200
290 U=1                                   -Initialize counter for factor
                                           power
300 N=N/K @ IF MOD(N,K)=0 THEN U=U+1 @    -Remove all possible factors K
    GOTO 300 ELSE 310
310 F=F+1 @ K(F)=K @ P(F)=U @ S=SQR(N)    -Update factor array with
                                           factor, power.  Revise S.
320 RETURN
330 IF N>1 THEN F=F+1 @ K(F)=N @ P(F)=1
340 DISP 'No.        Factor     Power'     -Display results
350 FOR I=1 TO F
360 DISP USING '2d,x,12d,5x,3d' ; I,K(I
    ),P(I)
370 A=FNA @ IF A=8 THEN I=MAX(I-1,1) @
    GOTO 360
```

PROGRAM LISTING

```
380 NEXT I
390 DISP CHR$(210);'un again, ';CHR$(21     -Continuation options
    4);'iew again, or ';CHR$(197);
400 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
410 ON POS('RVE',Q$)+1 GOTO 390,120,340
    ,420
420 DISP '           END OF PROGRAM' @ S
    TOP
```

NOTES

NOTES

MATH I

SIMULTANEOUS LINEAR EQUATIONS
QUADRATIC EQUATION
PARABOLIC EQUATIONS
ROOTS OF POLYNOMIALS
TRIANGLE SOLUTIONS
POLYGON AREA
HYPERBOLIC FUNCTIONS
COMPLEX TRIGONOMETRIC FUNCTIONS
PRIME FACTORIZATION

ALL HP-75 SOLUTIONS BOOKS ARE AVAILABLE RECORDED ON MINI-DATA CASSETTES FROM EITHER A HEWLETT-PACKARD DEALER OR THE HP USERS' LIBRARY.

00075-90061 Printed in U.S.A.

Reorder No.
00075-13003