SURVEYING PROGRAM SUITE
FOR TI84 FAMILY OF CALCULATORS
BY
JACK M. KESLER, RLS, USMS
11/30/2015
Version 1.00

INTRODUCTION:

This manual covers 17 surveying applications developed for the TI84 family of calculators. It is anticipated that the user is familiar with the stat, matrix and list editors. The mode should be set to degrees and decimals to 4. The applications depend on lists and matrices for primary data storage. The coordinate data base is matrix [C] and is used throughout the programs. By using TICONNECT™ this matrix may be exported to a CVS file which can be edited in EXCEL™ to add elevations and point descriptions and then loaded into a Cadd application for plotting.

By design these applications should address most requirements for handling surveys. The following lists must be created: PT, N, E, L, D, H, X, Y, PP, PN, AT, BS, and FS. These lists will reside in memory for all applications. Error management has been left to the compiler. The use of exact numbers of points, etc. must be adhered to so as to avoid problems. As the calculator has only a limited string variable function unless asked all input should be in numbers. If desired the user can construct a matrix containing all work lists which can be exported and copied using the command “List►matr(lists,matrix)”.

Angles and azimuths are maintained in decimal degrees and outputted to DMS when needed. The quadrant codes either input or output are: 1=NE, 2=SE, 3=SW and 4=NW. The beauty of this little device is the ability to edit data entry without redoing the application at the input stage. The programs allow these changes to be seamlessly integrated into the final results. This is particularly beneficial when using the Least Squares Programs! Coordinates may be imported using the computer program TICONNECT™ or may be entered in the matrix editor. A program has been added to aid in this process if needed. Note when the display pauses it must be continued by hitting enter. Please note that I sharing all of this information and all I ask is that I be given credit for these documents and the code.

Attached to this manual are resource papers which go into detail on the more complicated programs in the lists together with samples of the programs.

INDEX:

ADJAREA (Ver 2.0):

This program takes a coordinate file presumably created either by ANGTRA or AZBTRA and balances the coordinates using the Compass Rule, Transit Rule or Crandall Rule. It is designed to use the balancing rules on an open traverse which starts at a fixed point and azimuth and ends on a fixed point and azimuth or a loop traverse beginning at a fixed point and ending on the same point. If angles are used to compute azimuths the program ANGTRA allows an equal angle adjustment of misclosure in angles prior to computing the final coordinates.

The Main Menu:

ADJAREA

1. **NUM PTS**
2. **LOAD LIST**
3. **ADJUST**
4. **AREA**
5. **SAVE PTS**
6. **EXIT**

NUM PTS:
Screen text in order of appearance;

ENTER NO. PTS:
This routine requires entry of the number of points being used in the traverse including the closing point. For example a loop traverse containing 4 points would be entered as 5 points! The lists are then setup with a 4 point buffer. The lists required are: PT, N ,E, L, D, H, PP, PN and L1.

LOAD LIST:

This routine asks the following questions all of which must be answered.

ENTER POINTS:
This routine asks the user to enter the point numbers to be worked based on the previous value entered. These points may be any points in the data base as long they proceed in a Clockwise direction and create a continuous traverse. The program accesses these by using the list location PT.

TYPE TRAVERSE:
This routine asks the user which type of traverse is being worked either LOOP or OPEN. The answer then enters the Closing point number and gives the user an option to key in a new point not in the file or recall a point number. Once the data has been entered it then runs and outputs the following data: LENGTH FIG, ERR IN LAT, ERR IN DEP and REL ERROR AS 1: RELATIVE VALUE. If these values indicate a blunder in entry rerun to the LOAD LIST option.

ADJUST:
This menu item routes the user to a sub menu:

ADJ METHOD

1. **COMPASS RULE**
2. **TRANSIT RULE**
3. **CRANDALL RULE**
4. **EXIT**

The coordinates are then balanced using whichever rule chosen and stored in the lists. The values in the coordinate file are not changed and a menu item will direct this option.

AREA:
This item will compute the area based on the lists. If the user is trying to compute an area for an open traverse an error message is shown: NO AREA EXISTS. On the other hand if a loop traverse is being worked then the following output is shown: AREA SQ. FEET and AREA ACRES. These are not saved so write them down.

SAVE:
This item should be used only when the user is satisfied with the accuracy of the input and the results. It will overwrite the points indicated in the PTS list.

EXIT:
This item ends the program. If the points have not been saved it is not to late, as the lists still maintain all of the data and the user can restart the program and use SAVE.

WORK LIST L1:

L1(1), L1(2), L1(3) and L1(4) are used by the Crandall Rule. L1(5) = closure in Latitude and L1(6) = closure in Departure L1(7),L1(8) and L1(9) are also used by the Crandall Rule.

ANGTRA (Ver 1.3):

This program allows the user to enter a fixed azimuth or bearing and using angles RT or LT and deflection angles RT or LT to compute azimuths, adjust them using the equal angle rule and compute raw coordinates to be balanced in ADJAREA.

The Main Menu:

ANGTRA

1. **SETUP LISTS**
2. **ENTER BS-FS**
3. **TYPE TRAV**
4. **ADJ ANGLES**
5. **COMP TRAV**
6. **EXIT**

SETUP LISTS:

This routine sets up working lists: L1, L2, L3 and H. At this time the program allows for 20 legs only. The specific numbers of points and angles will be entered later. If more is required contact me at: jkesler@citlink.net.

ENTER BS-FS:

This routine asks for the BS bearing and asks if this bearing is toward the base point or away. The program requires the BS bearing to be away from the base point. The FS bearing is assumed to be away from the closing point.

TYPE TRAV:

This routine starts a sub menu:

TYPE TRAV

1. **NO PTS-ANGS**
2. **LOOP TRAV**
3. **OPEN TRAV**
4. **EXIT**

NO PTS-ANGS:

This routine must be accessed first and once the data is entered the user will return to this sub menu and must choose OPEN or LOOP. EXIT returns control to the main menu. When counting the number of points begin with the fixed first point and then proceed through the traverse to the closing point. For example a loop traverse containing 4 points would be entered as 4 the same with an open traverse. The number of angles includes beginning, each traverse angle and the closing angle.

LOOP AND OPEN:

These two routines access separate data entry routines for loop and open traverses separately. Choose only one!

These two routines route the user to a submenu:

TYPES OF ANG

1. **ANG RT**
2. **ANG LT**
3. **DEF RT**
4. **DEF LT**
5. **EXIT**

These sub routines allow the entry of the four basic types of angles and converts same to angles right for the number of angles indicated above. EXIT returns to the main menu. Returning to the Main menu.

ADJ ANGLES:

This routine then corrects each angle by equal angle corrections. The angular closure is displayed as is the correction per each angle, and a sum check which should be the closing bearing and the program pauses as before. When the user hits enter all angles are corrected. If the closure is not within reason it may be a bum angle entry and the sub routine should be re run until the user is happy with the results.

COMP TRAV:

This routine then computes corrected azimuths and asks for distance input and proceeds to compute the coordinates.

SAVE:

This routine then copies the new coordinates to the data file to be used in the ADJAREA program.

AZBTRA (Ver 2.0):

This program allows traversing either by bearings or azimuths. Provision is made to place a number of points on the same line by distance entry and by division into parts. A routine is added indicating the next free point ready for storage.

The Main Menu:

TRAVERSE

1. **BEG NEW CRDS**
2. **POB POINT**
3. **TRAVERSE**
4. **POL PTS**

5. **NEXT FREE PT**
6. **EXIT**

BEG NEW CRDS:

This routine allows the user to enter the coordinates for the beginning of a new traverse assuming the point is not in the data file.

POB POINT:

This routine allows for the beginning of a new traverse at an existing point in the data file.

TRAVERSE:

This routine does the traversing work and when accessed asks for the “AT PT” to start at and the point to compute “TO PT” and to exit enter “0” when asked for the next point which has been incremented and displayed then jumps to a sub menu:

AZI-BRG

1. **BEARING**
2. **AZIMUTH**
3. **EXIT**

BEARING:

This routine requires inputting a quadrant code and a bearing angle.

AZIMUTH:

This routine requires entering an azimuth as measured from North..

If it is desired to mix bearings and azimuths enter 0 for next point and re access the TRAVERSE routine.

POL PTS:

This routine allows the user to recall two points and traverse along the line. The distance is recalled and the user may input a factor to divide the distance into parts and compute new coordinates.

NEXT FREE PT:

This routine finds the next free point in the data file if needed.

EXIT:

This routine as always closes the program and returns to the Home screen.

HCURVE (Ver. 2.0):

This program allows the user to compute horizontal curves given two of the most frequent given parameters, fit a curve given two tangents and a radius, compute areas for the curve sector, segment and fillet, compute coordinates for the PC, PT and PR and save the curve data for later download to EXCEL™. These routines currently work with the arc definition for horizontal curves.

The Main Menu:

H CURVE SOLN

1. **SETUP LISTS**
2. **CRV DATA**
3. **CRV FIT**
4. **AREAS**
5. **PC PT PR**
6. **SAVE DATA**
7. **EXIT**

SETUP:

This routine sets up the following lists:

Working Lists:
L1(1) = Radius.
L1(2) = Degree of curve in decimal degrees.
L1(3) = Delta angle in decimal degrees.
L1(4) = Length of arc.
L1(5) = Tangent.
L1(6) = Chord (Long chord).
L1(7) = External.
L1(8) = Middle ordinate.
L1(9) = FS azimuth in decimal degrees.
L1(10) = Deflection angle at PI in decimal degrees.
L1(11) = BS azimuth in decimal degrees.
L1(12) = 5729.57795 (Radius of a one degree curve = 180º/π).
L1(13) = At point number.
L1(14) = BS point number.
L1(15) = FS point number.
L1(16) = Sector area in square feet.
L1(17) = Segment area in square feet.
L1(18) = Fillet area in square feet.
L1(19) = Curve number.

Storage Lists:
L2() = Delta angles in decimal degrees.
L3() = Radius.
L4() = Length of arc.
L5() = Tangent.
L6() = Chord.
LPT = Curve number.
LN = External.
LE = Middle ordinate.

LAT = Sector area.
LBS = Segment area.
LFS = Fillet area.

The following Lists are to be converted to matrix [A] in this order (using List►Matrix):

PT,L2,L3,L4,L5,L6,N,E,AT,BS,FS,[A]

CRV DATA:

This routine routes the user to a sub menu:

KNOWN PARMS

1. **R-T**
2. **C-R**
3. **L-R**
4. **Θ-R**
5. **T-C**
6. **EXIT**

Note θ = delta angle.

The terms are self explanatory namely with the known items the rest of the curve elements are determined and the areas may be computed. The PC PT PR points are not computed.

CRV FIT:

This routine requires a PI point number (assumed to be in the data file), a Back sight number (assumed to be in the data file), a Fore sight number assumed to be in the data file) and a radius. All curve elements are computed.

AREAS:

This routine takes the current curve data stored in L1 and computes the sector, segment and fillet areas.

PC PT PR:

This routine takes the FIT CRV data and computes the coordinates for the PC, PT and PR and stores them in the data file using the next free coordinates.

SAVE DATA:

This routine takes the current curve solutions and stores them in the next free Lists if desired. If the user does not wish to save the curve data then do not activate the routine.

Each routine pauses and waits for input as asked. EXIT stops the programs and either returns to the Main Menu or to the Home page.

INTRSCT (Ver. 2.0):

This program solves the three primary COGO routines namely Bearing to Bearing, Bearing to Distance and Distance to Distance.

The Main Menu:

INTERSECT

1. **SETUP LIST**
2. **BB**
3. **BD**
4. **DD**
5. **EXIT**

SETUP LIST:

This routine clears and creates the work list and must be used before using the solution routines:

For Bearing to Bearing:

L1(1) = Base line azimuth determined by inversing and in decimal degrees. Base line distance is also computed and stored in ”H”.
L1(2) = Azimuth from Base point “A” to unknown point “P”.
L1(3) = Azimuth from Base point “B” to unknown point “P”.
L1(4) = L1(1) – L1(2).
L1(5) = L1(3) – L1(1).
L1(6) = 180 – [L1(4) + L1(5)] which is the third angle in the triangle A-P-B-A.

For Bearing to Distance:

L1(1) = Base line azimuth determined by inversing and in decimal degrees. Base line distance is also computed and stored in “H”.
L1(2) = Azimuth from Base point “A” to unknown point “P”.
L1(3) = absolute value [L1(1) – L1(2)] and modified to be the angle from point “P” to Base point “B”.
L1(4) = Distance to intersection point.
L1(5) = Test distance computed at right angles from Base point “B” to the line A-P. If this value is greater than L1(4) then no solution exists.
L1(6) = Computed distance on each side of the point on line A-P from point “B”.

For Distance to Distance:

L1(1) = Base line azimuth determined by inversing and in decimal degrees. Base line distance is also computed and stored in ”H”.
L1(2) = Distance from A-P.
L1(3) = Distance from B-P.
If L1(2) + L1(3) is less than “H” no solution exists.

BB (Bearing to Bearing):

This routine uses two bearings either entered or computed to compute the position of the unknown point “P”. The RULE: pick Base Point “A” so that the solution can proceed clockwise A to P to B and enter data accordingly. For this routine only one solution exists.

BD (Bearing to Distance):

This routine uses a bearing and a known distance from the base line to compute two solutions. The RULE: pick Base point “A” to be root of known bearing nearest the point to be computed and Base point “B” to be root of the known distance. If point “A” is chosen as suggested then the first solution is always the one desired. The second solution will be saved and can be deleted in the Matrix Editor if desired.

DD (Distance to Distance):
This routine uses two distances from the base line to compute two solutions. The RULE pick Base point “A” so that the solution can proceed in a clockwise direction and if so the first point computed is always the one desired. The second solution will be saved and can be deleted in the Matrix Editor if desired.

And of course EXIT stops the program and returns to the Home page.

INVERSE (Ver. 3.0):

This program allows the user to compute the bearings and distances between points in the data file and the angle right subtended by two lines as measured at the apex.

The Main Menu:

INVERSE

1. **SETUP**
2. **INVERSE**
3. **COMP ANG**
4. **EXIT**

SETUP:

This routine will setup the work list for the COMP ANG routine:

Work List:

L1(1) = PI point number.
L1(2) = BS point number.
L1(3) = FS point number.

INVERSE:

This routine takes inverses in bearing and distance between thw first point and the second point and will continue until “0” is entered as the point number.

COMP ANG:

This routine uses points in the data list to compute the angle to the right at PI from BS to FS.

EXIT stops the program.

LSDADJ (Ver. 2.0):

This program adjusts an open traverse (beginning at a known point with a fixed azimuth and closing on a known point with a fixed azimuth) by the Condition Equation approach designed by Joseph Dracup late of the USC&GS. To use this program first generate measured angles, HD and N and E in the ANGTRA program. DO NOT ClrAllLists!!!

The Main Menu:

LSTADJ

1. **SETUP**
2. **DIAG OF [P]**
3. **ROW 1-2-3 [B]**
4. **COL 1 [D]**
5. **SOLVE ADJ**
6. **SAVE CRDS**
7. **EXIT**

SETUP:

Work List:

L1(1) = BS azimuth at POB point in decimal degrees.
L1(2) = FS azimuth at POE point in decimal degrees.
L1(3) = Number of angles.
L1(4) = Number of distances.
L1(5) = Number of points.
L1(6) = 206264.81 seconds per radian.
L1(7) = POE Northing.
L1(8) = POE Easting.
L1(9) = Sum of angles + BS azimuth in decimal degrees. [SUM(L2,1,L1(5))+L1(1)] store to L1(9) = closure in angles in decimal degrees once the number of 180 degrees have been adjusted.
Some of L1() is carried over from ANGTRA program.
L2() = angles carried over from the ANGTRA program.
LN() = Northings carried over from ANGTRA program.

LE() = Eastings carried over from ANGTRA program.

Output Matrices:

[A]() = dim(T,1) which is the correction or answer matrix.
[B]() = dim(3,T) which is the design matrix.
[D]() = dim(3,1) which is the closure value matrix.
[D](1,1) = Az(computed) – Az(fixed) in arc seconds.
[D](2,1) = 206264.81*(X(computed) – X(fixed)) in arc seconds.
[D](3,1) = 206264.81*(Y(computed) – Y(fixed)) in arc seconds.
Identity(T) store to [E]. which is the weight matrix.
Where T = L1(3) + L1(4).

DIAG OF [P]:

This routine loads weights into the [E] matrix if a weighted solution is to be used. The values are computed by entering the a'priori standard errors. If it is desired not to use weights skip this routine

ROW 1-2-3:

This routine computes the rows for the design matrix.

COL 1 [D]:

This routine loads the closure values into the [D] matrix.

SOLVE ADJ:

This routine solves for either a weighted or un-weighted solution.

SAVE CRDS:

This routine saves the adjusted coordinates to the data file.

And EXIT ends the program. No error analysis a'posteriori is determined. This program and the LSWADJ provide identical results with the same SETUP routine but the approach to weighting is different. In LSWADJ the user must physically enter "1" for each weight matrix element to achieve an un-weighted solution. LSDADJ uses arc seconds and LSWADJ uses radians. Both of these programs have been kept because of the effort in producing them together with a positive means of checking the solutions.

LSPADJ (Ver. 1.1):

This program uses the power of the TI84 family to solve matrices in solving the parametric least squares solution based on observation equations for angles, distances and bearings. The solution is weighted and an error analysis is provided using the LSPERR program immediately after making the adjustment

The Main Menu:

OBS EQNS

1. **DESIGN**
2. **O-C VECTOR**
3. **SOLVE**
4. **UPDATE**
5. **SAVE**
6. **EXIT**

DESIGN:

This routine branches to a sub menu:

DESIGN INPUT

1. **SETUP**
2. **ANGLES**
3. **DISTANCES**
4. **BEARINGS**
5. **EXIT**

SETUP:

This routine sets up the work list:

Lists:
L1(1) = Number of points.
L1(2) = Number of angles.
L1(3) = Number of distances.
L1(4) = Number of free points.
L1(5) = At point number.
L1(6) = From point number
L1(7) = to point number.
L1(8) = 206264.81 secs/radian = ρ
L1(9) = Number of bearings.
L1(10) = Temp At to FS azimuth.
L1(11) = Temp At to BS azimuth.
L2() = Angles in decimal degrees.
L3() = Azimuths in decimal degrees.
L4() = Corrections.
LPT(), LX(), LY(), LH(), LAT(), LBS() and LFS().

Matrices:
[A]() = Correction or answer matrix
[B]() = Design matrix.
[E]() = Weight matrix.
[D]() = O-C matrix.

ANGLES:

This routine branches to a sub menu:

ANGLES

1. **OBS ANG**
2. **OBS COEF**
3. **EXIT**

OBS ANG:

This routine prompts for the measured angles and each angles a'priori standard error.

OBS COEF:

This routine generates the observation equation coefficients for each angle (keep them in order). The user is prompted for the AT , BS and FS point numbers. Fixed points are negated. EXIT returns to the DESIGN menu.

DISTANCES:

This routine branches to a submenu:

DISTS

1. **OBS DIST**
2. **OBS COEF**
3. **EXIT**

OBS DIST:

This routine prompts the user for the measured distances and each distances a'priori standard error.

OBS COEF:

This routine generates the observation equation coefficients for each distance (keep them in order). The user is prompted for the AT, FS point numbers. Fixed are negated. EXIT returns to the DESIGN menu.

BEARINGS:

This routine branches to a submenu:

BEARINGS

1. **OBS BRG**
2. **OBS COEF**
3. **EXIT**

OBS BRG:

This routine prompts the user for the measured bearings and each bearings a'prior standard error.

OBS COEF:

This routine generates the observation equation coefficients for each bearing (keep them in order). The user is prompted for the At, FS point numbers. Fixed are negated. EXIT returns ro the DESIGN menu.

EXIT returns to the main menu.

O-C VECTOR:

This routine generates the Observed minus computed values.

SOLVE:

This routine solves the resulting adjustment using Matrix algebra and displays all of the corrections for DX and DY. The variance/covariance matrix is computed and stored in [F]().

UPDATE:

This routine updates the preliminary coordinates and re running the O-C VECTOR and SOLVE the user can iterate the solution until the corrections are small enough (Usually 0.001' or less).

SAVE:

This routine takes the final coordinate values and saves them to the data file.

EXIT:

This routine ends the program though leaving all Lists and Matrices intact.

LSPERR (Ver. 1.1):

This program is a continuation of LSPADJ and contains routines to compute the standard error a'posteriori of the coordinates and the 95% absolute error ellipses. These items are stored in lists and can be converted to a matrix and opened in EXCEL by using the tools in TI-CONNECT.

The Main Menu:

LSPERRS

1. **SETUP LISTS**
2. **SE OF UNIT WEIGHT**
3. **LOAD Q DATA**
4. **POINTS ERR**
5. **ERR ELLIPS**
6. **EXIT**

SETUP LISTS:

The following three matrices stored by LSPADJ are required for use by this program. They may still reside in memory or can be loaded from a computer at a later time:

[B]() = design matrix.
[D]() = residual matrix.
[E]() = weight matrix.
[F]() = variance/covariance matrix.

The matrix [I](1,1) is created to store the standard error of unit weight which is then stored in list item L1(6). This is because of TI84 storing rules.

Work Lists:

L1(1) = number of free points.
L1(2) = number of unknowns (number of points times 2).
L1(3) = number of observations.
L1(4) = degrees of freedom.
L1(5) = 1.96 which is the 95% probability multiplier.
L1(6) = standard error of unit weighr.
L1(7) = QXY.
L1(8) = QXX.
L1(9) = QYY.
L2() = X standard error.
L3() = Y standard error.
L4() = semi-major axis 95%.
L5() = semi-minor axis 95%.
L6() = azimuth of semi-major axis.

SE OF UNIT WT:
This routine computes the standard error of unit weight using the matrices: [D] and [E].

LOAD Q DATA:
This routine uses the [F] matrix and computes and stores the standard error of the coordinates and 95% error ellipse parameters.

POINTS ERR:
This routine runs through the point standard errors on the home screen.

ERR ELLIPS:
This routine runs through the error ellipse parameters on the home screen.

EXIT:
This routine as always ends the program.

PTAREA (Ver. 1.0):

This program uses the data file to compute the area of a closed traverse using point numbers entered by the user. If the closed traverse has 4 points then the program needs one additional point namely the beginning point when asked for the number of points. If curves are involved traverse the chord or the PI and then compute the curve.

The Main Menu:

AREA BY POINTS

1. **SETUP LISTS**
2. **ENTER PTS**
3. **COMPUTE**
4. **EXIT**

SETUP LISTS:

This routine creates the eork lists and asks for the number of points:

Work Lists:

LPT() = point numbers.
LN() = N coordinate for each point.
LE() = E coordinate for each point.
LPP() = positive areas.
LPN() = negative areas.

ENTER PTS:

This routine prompts for the points making up the figure whose area is required. Points entered clockwise with the beginning point entered first and last.

COMPUTE:

This routine computes the area in square feet and acres.

EXIT:

This routine of courses ends the program.

RADIAL (Ver. 1.0):

This program allows the user to compute side shots while located at a fixed point by angles measured from a known point or fixed azimuth or use bearings or azimuths.

The Main Menu:

RADIAL

1. **SETUP**
2. **BASE PT**

3. **REF AZ**
4. **ANG COMP**
5. **BRG COMP**
6. **AZI COMP**
7. **EXIT**

SETUP:

This routine creates the work lists:

Work Lists:

L1(1) = BS azimuth in decimal degrees.
L1(2) = N coordinate of base point.
L1(3) = E coordinate of base point.

BASE PT:

This routine enters the base point number.

REF AZI:

This routine branches to a submenu:

BS BRG

1. **ENTER**
2. **COMP**
3. **EXIT**

ENTER:

This routine allows the user to key in the BS bearing.

COMP:

This routine allows the BS bearing to be computed from another point.

EXIT;

This routine returns to the main menu.

ANG COMP:

This routine branches to a sub menu:

TYPES OF ANG

1. **ANG RT**
2. **ANG LT**
3. **DEF RT**
4. **DEF LT**
5. **EXIT**

ANG RT:

This routine allows entering an angle to the right.

ANG LT:
This routine allows entering an angle to the left.

DEF RT:
This routine allows entering a deflection angle to the right from the BS produced.

DEF LT:
This routine allows entering a deflection angle to the left from the BS produced.

EXIT:
This routine returns to the main menu.

BRG COMP:
This routine allows using a quadrant code and a bearing angle.

AZI COMP:
This routine allows using an azimuth as measured from North.

EXIT:
This routine of course ends the program.

RESC3PT {Ver. 1.0):

This program computes coordinates of an unknown point at which angles have been measured between three known points.

The Main Menu:

3 POINT

1. **SETUP**
2. **ENTER PTS**
3. **ENTER ANGS**
4. **SET CNTRL**
5. **COMPUTE**
6. **EXIT**

SETUP:
This routine sets up the work lists:

Work Lists:
L1(1) = Angle 1 (A).
L1(2) = Angle 2 (B).
L1(3) = Angle 3 (C).
L1(4) = Azimuth 1.

L1(5) = Azimuth 2.
L1(6) = Back azimuth.
L1(7) = K.
L1(8) = Angles D + E.
L1(9) = TAN((D-E)/2).
L1(10) = (D-E)/2.
L1(11) = Angle E.
L1(12) = Angle D.
L1(13) = Azimuth to unknown point P.
L1(14) = Length L3.
The following lists are also used.
LPT()
LX()
LY()
LH()

ENTER PTS:

This routine enters the three fixed points. The program uses points 1 to 3 in order and as such the point numbers from the data base must be entered in a clockwise direction.

ENTER ANGS:

This routine enters the two measured angles which must be entered in a clockwise direction.

SET CNTRL:

This routine uses inverses between the three fixed points and computes the Angle C.

COMPUTE:

This routine solves the three point problem and allows storing the coordinates of point P.

EXIT:

This routine as always ends the program.

As sound professional work requires redundancy to verify the solution then if other fixed points are observed numerous solutions can be computed and stored. A short program entitled SUMPTS is included which allows summing and finding the mean coordinates for the unknown point. The program LSPADJ will allow computing a least squares solution as well. These methods and samples will be addressed in the resource paper.

SCROTR (Ver. 1.0):

This program allows the user to scale, rotate and translate one coordinate system (the old) to a new coordinate system.

The Main Menu:

SCROTR

1. **ENTER DATA**
2. **SCALE**
3. **ROTATION**
4. **TRANSLATE**
5. **COMPUTE**
6. **SAVE PTS**
7. **EXIT**

ENTER DATA:

This routine requires inputting the new coordinates for the “pivot point”. An additional point may be entered if two points are known in the old and new systems. This allows computing the scale and rotation directly. As noted enter “0” if none.

SCALE:

This requires entering the scale factor or computing from two points. It branches to a sub menu:

SCALE OPTIONS

1. **ENTER K**
2. **COMPUTE K**
3. **EXIT**

ENTER K:

This routine allows entering yhe scale factor or a unit conversion factor.

COMPUTE K:

This routine allows a scale factor to be computed from two points in each system. It is suggested that if two points are used the second point should be as far from the base pivot point as possible.

EXIT:

This routine returns to the Main Menu.

ROTATION:

This routine requires entering the rotation angle and branches to a sub menu:

NEW AZIMUTH

1. **RO ANGLE EXISTS**
2. **NEW BRG**
3. **COMPUTE**

4. EXIT

RO ANGLE EXISTS:
If a rotation exists (to the right) then it can be entered here.

NEW BRG:
If a new bearing exists it may be entered at this point.

COMPUTE:
This routine allows computing the angle from the 2 points in each system.

EXIT:
This routine returns to the main menu.

TRANSLATE:
This routine allows the old system to be translated to the new system. It is important to note that the scale and rotation routines must be run first before translating as the factors depend on the rotation angle. If no scale exists then enter K = 1. If no rotation exists then enter the RO angle as “0”.

COMPUTE:
This routine computes the final new coordinates. The new values are stored in Lists:

Work Lists:
L2() = Point numbers.
L3() = Northings.
L4() = Eastings.

The entire setup is amplified in the resource paper.

SAVE:
This routine saves the coordinates to the data file.

EXIT:
As always ends the program.

SDREDUC (Ver. 2.0):

This program takes an HIT, HIEDM, HP, HS, Zenith angle and Slope distance to be reduced to a distance at the lowest station corrected for curvature and curvature & refraction. A routine for determining the Mark to Mark slope distance and Mark to Mark zenith angle is included should the user desire to incorporate the length into a GPS network.

The Main Menu:

SD REDUC

1. **SETUP LIST**
2. **SDRED**
3. **MTOM**
4. **EXIT**

SETUP LIST:

This routine clears and creates L1() as shown in the resource paper.

SDREDU:

This routine requests data and reduces same to a horizontal distance at the lower point and an elevation difference between the control points.

MTOM:

This routine computes the Mark to Mark Slope distance and zenith angle between the monuments on the earth's surface.

EXIT:

This routine of course ends the program.

SETCRDF (Ver. 2.0):

This utility program sets up a new data file, fills it with "0's", clears all lists and allows the manual entry of coordinates.

The Main Menu:

FILE APPS

1. **SETUP FILE**
2. **CLEAR FILE**
3. **INPUT CRDS**
4. **CLEAR LISTS**
5. **EXIT**

SETUP FILE:

This routine creates a new [C] matrix.

CLEAR FILE:

This routine fills the new or existing [C] matrix with "0's".

INPUT CRDS:

This routine allows the hand entry of coordinates. This activity can be avoided by putting a coordinate file in to EXCEL™ saving same to *.csv file and using TI CONNECT to import this file in to a calculator file and same loaded into the TI84

CLEAR LISTS:

This routine allows the user to clear all values from the Lists. Most of the programs in this suite will re-diminsion the lists required.

EXIT:

This routine of course ends the program.

SUMPTS (Ver. 1.0):

This program is meant to work with the three point program and allows the summing of numerous solutions for the same point, finding the mean and saving this mean to the data file.

The Main Menu:

SUM/MEAN

1. **NO: OF PTS**
2. **SUMS**
3. **MEANS**
4. **SAVE MEANS**
5. **EXIT**

NO: OF PTS:

This routine requests the beginning at point and the go to point numbers.

SUMS:

This routine then computes the sum of points.

MEANS:

This routine computes the means.

SAVE MEANS:

This routine saves the mean coordinates.

EXIT:

This routine of course ends the program.