```
' Program CO2SYS.BAS, version 01.05, 10-15-97, written by Ernie Lewis.
' This is a new version combining CO2SYSTM, FCO2TCO2, PHTCO2, and CO2BTCH.
' For more information, see the sub AboutCO2SYS.
'
'
' ***********************************************
' !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
' PROGRAMMER'S NOTE: This program is DANGEROUSLY close to the DOS-imposed
'       64K limit due to all the print statements in the sub AboutCO2SYS.
'       Don't make any unnecessary changes or the limit will be exceeded.
' !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
' PROGRAMMER'S NOTE: all logs are base e, any log10 is written log()/log(10)
' PROGRAMMER'S NOTE: all temps are deg C unless otherwise noted -
'       temps in deg K only occur in the subs and are expicitly noted
' PROGRAMMER'S NOTE: partials are calculated numerically and there will be
'       some roundoff error involved in this, but it should be small
' PROGRAMMER'S NOTE: pCO2 and fCO2 are both referenced to wet air. In an
'       earlier version I had xCO2 in dry air as a variable with pTot
'       assumed to be 1 atm (so essentially I had pCO2 in dry air), thus
'       there is some code that could be removed now if I chose to do so.
'       FugFac does not change with TempC very much, whereas VPFac = (1-pH2O)
'       did, so I could put it as a constant, but I left the code as it was.
' PROGRAMMER'S NOTE: the constants are converted to the chosen pH scale and
'       calculations are made on that scale. Some of the subs are designed
'       for the total scale, but for reasonable pH (>6) they will work fine.
' PROGRAMMER'S NOTE: the statement:
'       IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
'       is so the TA value used is the correct one for the case used (Peng or
'       Dickson); the program is coded for TA(Dickson) in the calculation subs.
'
'
'****************************************************************************
DeclareStuff:
        DECLARE SUB AboutCO2SYS (Info$)
        DECLARE SUB CalculateAlkParts (pH, TC, K(), T(), HCO3, CO3, BAlk, OH,
PAlk, SiAlk, Hfree, HSO4, HF)
        DECLARE SUB CalculatefCO2fromTCpH (TC, pH, K0, K1, K2, fCO2)
        DECLARE SUB CalculatepHfromTAfCO2 (TA, fCO2, K0, K(), T(), pH)
        DECLARE SUB CalculatepHfromTATC (TA, TC, K(), T(), pH)
        DECLARE SUB CalculatepHfromTCfCO2 (TC, fCO2, K0, K1, K2, pH)
        DECLARE SUB CalculateTAfromTCpH (TC, pH, K(), T(), TA)
        DECLARE SUB CalculateTCfrompHfCO2 (pH, fCO2, K0, K1, K2, TC)
        DECLARE SUB CalculateTCfromTApH (TA, pH, K(), T(), TC)
        DECLARE SUB Case1Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TA, TC, pHinp, fCO2inp,
pCO2inp, pHout, fCO2out, pCO2out)
        DECLARE SUB Case2Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TA, pHinp, TC, fCO2inp,
pCO2inp, pHout, fCO2out, pCO2out)
        DECLARE SUB Case3Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TA, fCO2inp, pCO2inp, TC,
pHinp, pHout, fCO2out, pCO2out)
        DECLARE SUB Case4Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TC, pHinp, TA, fCO2inp,
pCO2inp, pHout, fCO2out, pCO2out)
```

```
        DECLARE SUB Case5Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TC, fCO2inp, pCO2inp, TA,
pHinp, pHout, fCO2out, pCO2out, TCfCO2Flag%)
        DECLARE SUB Case6Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, pHinp, fCO2inp, pCO2inp, TA,
TC, pHout, fCO2out, pCO2out)
        DECLARE SUB CaSolubility (WhichKs%, Sal, TempC, Pdbar, TC, pH, K1, K2,
OmegaCa, OmegaAr)
        DECLARE SUB ChoosefORp (fORp$)
        DECLARE SUB ChooseInputMode (Batch$)
        DECLARE SUB ChoosepHScale (pHScale$)
        DECLARE SUB ChooseWhichCase (ICase%, fORp$)
        DECLARE SUB ChooseWhichKs (WhichKs%, pHScale$)
        DECLARE SUB ChooseWhoseKSO4 (WhoseKSO4%)
        DECLARE SUB Constants (pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC,
Pdbar, K0, K(), T(), fH, FugFac, VPFac)
        DECLARE SUB ErrorSub (IError%, Error$)
        DECLARE SUB FindpHfCO2fromTATC (pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
        DECLARE SUB FindpHOnAllScales (pHScale$, pH, K(), T(), fH, pHNBS,
pHfree, pHtot, pHsws)
        DECLARE SUB InputfCO2 (fCO2)
        DECLARE SUB InputFileInfo (NHeaderLines%, NIDFields%, MVD, InputFile$,
OutputFile$, MVFlag$)
        DECLARE SUB InputFromBatchFile (NIDFields%, MVD, MVFlag$, BatchFlag%,
Sal, TP, TSi, TempCinp, Pdbarinp, TempCout, Pdbarout, Param1, Param2, MVFlag)
        DECLARE SUB InputParameters (WhichKs%, Sal, TP, TSi, TempCinp, Pdbarinp,
TempCout, Pdbarout)
        DECLARE SUB InputpCO2 (pCO2)
        DECLARE SUB InputpH (pHScale$, pH)
        DECLARE SUB InputTA (TA)
        DECLARE SUB InputTC (TC)
        DECLARE SUB PrintFirstScreen ()
        DECLARE SUB PrintfpCO2OmegasRevelle (WhichKs%, fCO2inp, pCO2inp,
xCO2dryinp, Revelleinp, OmegaCainp, OmegaArinp, fCO2out, pCO2out, xCO2dryout,
Revelleout, OmegaCaout, OmegaArout)
        DECLARE SUB PrintHeader (ICase%, pHScale$, fORp$, TA, TC, pHinp,
fCO2inp, pCO2inp, TP, TSi, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout)
        DECLARE SUB PrintHeaderOnOutputFile (WhichKs%, WhoseKSO4%, fORp$,
pHScale$, ICase%, InputFile$, OutputFile$, NIDFields%, MVFlag$)
        DECLARE SUB PrintInputChoices (WhichKs%, WhoseKSO4%, fORp$, pHScale$,
Batch$)
        DECLARE SUB PrintKComments (WhichKs%)
        DECLARE SUB PrintpHspKs (pHScale$, WhichKs%, WhoseKSO4%, TA, TC, Sal,
K(), T(), TempCinp, Pdbarinp, TempCout, Pdbarout)
        DECLARE SUB PrintSpeciation (WhichKs%, TA, TC, HCO3inp, CO3inp, BAlkinp,
OHinp, PAlkinp, SiAlkinp, HCO3out, CO3out, BAlkout, OHout, PAlkout, SiAlkout)
        DECLARE SUB PrintTCfCO2Warning ()
        DECLARE SUB RevelleFactor (WhichKs%, TA, TC, K0, K(), T(), Revelle)
        DECLARE SUB SetDefaultsForCO2SYS (ICase%, WhichKs%, WhoseKSO4%, fORp$,
pHScale$, Batch$, TA, TC, pHinp, fCO2inp, pCO2inp, Sal, TempCinp, Pdbarinp,
TempCout, Pdbarout, NHeaderLines%, NIDFields%, MVD, MVFlag$)
        DECLARE SUB SetInputChoices (WhichKs%, WhoseKSO4%, fORp$, pHScale$,
Batch$)
        DECLARE SUB SetParametersForPartials (dTA, dTC, dpH, dfCO2, dSal,
dTempC, dPdbar, pcdK0, pcdK1, pcdK2)
'
```

```
'
DimensionStuff:
        DIM K(10): ' these are the equilibrium constants
        DIM T(5):  ' these are the amounts of the various species
'
'
'*****************************************************************************
'*****************************************************************************
        ON ERROR GOTO ErrorHandler:
Top:
        CALL PrintFirstScreen
        Info$ = "General": CALL AboutCO2SYS(Info$)
        CALL SetDefaultsForCO2SYS(ICase%, WhichKs%, WhoseKSO4%, fORp$, pHScale$,
Batch$, TA, TC, pHinp, fCO2inp, pCO2inp, Sal, TempCinp, Pdbarinp, TempCout,
Pdbarout, NHeaderLines%, NIDFields%, MVD, MVFlag$)
Start:
        pH = pHinp: fCO2 = fCO2inp: pCO2 = pCO2inp
        WhereFlag% = 0: ' this is a flag for help in error location
        CALL SetInputChoices(WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$)
        CALL ChooseWhichCase(ICase%, fORp$)
        IF Batch$ = "YES" THEN
                CALL InputFileInfo(NHeaderLines%, NIDFields%, MVD, InputFile$,
OutputFile$, MVFlag$)
OpenFiles:
                WhereFlag% = 1: ' this is a flag for help in error location
                OPEN InputFile$ FOR INPUT AS #3
                FOR I% = 1 TO NHeaderLines%
                        LINE INPUT #3, HeaderLine$
                NEXT I%
                WhereFlag% = 2: ' this is a flag for help in error location
                OPEN OutputFile$ FOR OUTPUT AS #4
                CALL PrintHeaderOnOutputFile(WhichKs%, WhoseKSO4%, fORp$,
pHScale$, ICase%, InputFile$, OutputFile$, NIDFields%, MVFlag$)
                NRead% = 1
InputHereForBatchMode:
                WhereFlag% = 3: ' this is a flag for help in error location
                IF EOF(3) = 0 THEN
                        BatchFlag% = 0
                        CALL InputFromBatchFile(NIDFields%, MVD, MVFlag$,
BatchFlag%, Sal, TP, TSi, TempCinp, Pdbarinp, TempCout, Pdbarout, Param1,
Param2, MVFlag)
                        NRead% = NRead% + 1
                        IF BatchFlag% = 1 THEN GOTO InputHereForBatchMode:
                        ' this is for end of file check and missing values stuff
                        IF WhichKs% = 6 THEN TP = 0!: TSi = 0!
                ELSE
                        CLOSE #3
                        CLOSE #4
                        DO: LOOP WHILE INKEY$ <> "": 'This clears the key
buffer.
                        LOCATE 23, 1: PRINT SPACE$(80): PRINT SPACE$(80);
                        LOCATE 24, 1
                        INPUT "    ANOTHER? (Y): ", Q$
                        IF LEN(Q$) <> 0 AND Q$ <> "y" AND Q$ <> "Y" THEN CLS :
END
                        GOTO Start:
                END IF
        ELSE
                CALL InputParameters(WhichKs%, Sal, TP, TSi, TempCinp, Pdbarinp,
TempCout, Pdbarout)
        END IF
        WhereFlag% = 4: ' this is a flag for help in error location
        T(4) = TP: T(5) = TSi
'****************************************************************************
InputCO2SystemParametersHere:
'       this is needed here for FugFac in converting between fCO2 and pCO2:
        TempC = TempCinp: Pdbar = Pdbarinp
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
        SELECT CASE Batch$
        CASE "YES"
                SELECT CASE ICase%
                CASE 1: ' input TA, TC
                        TA = Param1 * .000001
                        TC = Param2 * .000001
                CASE 2: ' input TA, pH
                        TA = Param1 * .000001
                        pH = Param2
                CASE 3: ' input TA, fCO2 or pCO2
                        TA = Param1 * .000001
                        IF fORp$ = "f" THEN fCO2 = Param2 * .000001: pCO2 = fCO2
/ FugFac
                        IF fORp$ = "p" THEN pCO2 = Param2 * .000001: fCO2 = pCO2
* FugFac
                CASE 4: ' input TC, pH
                        TC = Param1 * .000001
                        pH = Param2
                CASE 5: ' input TC, fCO2 or pCO2
                        TC = Param1 * .000001
                        IF fORp$ = "f" THEN fCO2 = Param2 * .000001: pCO2 = fCO2
/ FugFac
                        IF fORp$ = "p" THEN pCO2 = Param2 * .000001: fCO2 = pCO2
* FugFac
                CASE 6: ' input pH, fCO2 or pCO2
                        pH = Param1
                        IF fORp$ = "f" THEN fCO2 = Param2 * .000001: pCO2 = fCO2
/ FugFac
                        IF fORp$ = "p" THEN pCO2 = Param2 * .000001: fCO2 = pCO2
* FugFac
                END SELECT
        CASE ELSE
                SELECT CASE ICase%
                CASE 1: ' input TA, TC
                        CALL InputTA(TA)
                        CALL InputTC(TC)
                CASE 2: ' input TA, pH
                        CALL InputTA(TA)
                        CALL InputpH(pHScale$, pH)
                CASE 3: ' input TA, fCO2 or pCO2
                        CALL InputTA(TA)
                        IF fORp$ = "f" THEN CALL InputfCO2(fCO2): pCO2 = fCO2 /
FugFac
```

```
                        IF fORp$ = "p" THEN CALL InputpCO2(pCO2): fCO2 = pCO2 *
FugFac
                CASE 4: ' input TC, pH
                        CALL InputTC(TC)
                        CALL InputpH(pHScale$, pH)
                CASE 5: ' input TC, fCO2 or pCO2
                        CALL InputTC(TC)
                        IF fORp$ = "f" THEN CALL InputfCO2(fCO2): pCO2 = fCO2 /
FugFac
                        IF fORp$ = "p" THEN CALL InputpCO2(pCO2): fCO2 = pCO2 *
FugFac
                CASE 6: ' input pH, fCO2 or pCO2
                        CALL InputpH(pHScale$, pH)
                        IF fORp$ = "f" THEN CALL InputfCO2(fCO2): pCO2 = fCO2 /
FugFac
                        IF fORp$ = "p" THEN CALL InputpCO2(pCO2): fCO2 = pCO2 *
FugFac
                END SELECT
        END SELECT
'
'
'****************************************************************************
CalculateOtherParamsAtInputConditions:
        SELECT CASE ICase%
        CASE 1: ' input TA, TC
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                IF Batch$ = "YES" THEN
                        Param3 = pH
                        IF fORp$ = "f" THEN Param4 = fCO2 * 1000000!
                        IF fORp$ = "p" THEN Param4 = pCO2 * 1000000!
                        PRINT #4, USING " #.####, ####.#, "; Param3; Param4;
                        PRINT USING " #.####, ####.#, "; Param3; Param4;
                END IF
        CASE 2: ' input TA, pH
                IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                CALL CalculateTCfromTApH(TA, pH, K(), T(), TC)
                IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 =
fCO2 / FugFac
                IF Batch$ = "YES" THEN
                        Param3 = TC * 1000000!
                        IF fORp$ = "f" THEN Param4 = fCO2 * 1000000!
                        IF fORp$ = "p" THEN Param4 = pCO2 * 1000000!
                        PRINT #4, USING " ####.#, ####.#, "; Param3; Param4;
                        PRINT USING " ####.#, ####.#, "; Param3; Param4;
                END IF
        CASE 3: ' input TA, fCO2 or pCO2
                IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                CALL CalculatepHfromTAfCO2(TA, fCO2, K0, K(), T(), pH)
                CALL CalculateTCfromTApH(TA, pH, K(), T(), TC)
                IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                IF Batch$ = "YES" THEN
```

```
                        Param3 = TC * 1000000!
                        Param4 = pH
                        PRINT #4, USING " ####.#, #.####, "; Param3; Param4;
                        PRINT USING " ####.#, #.####, "; Param3; Param4;
                END IF
        CASE 4: ' input TC, pH
                CALL CalculateTAfromTCpH(TC, pH, K(), T(), TA)
                IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 =
fCO2 / FugFac
                IF Batch$ = "YES" THEN
                        Param3 = TA * 1000000!
                        IF fORp$ = "f" THEN Param4 = fCO2 * 1000000!
                        IF fORp$ = "p" THEN Param4 = pCO2 * 1000000!
                        PRINT #4, USING " ####.#, ####.#, "; Param3; Param4;
                        PRINT USING " ####.#, ####.#, "; Param3; Param4;
                END IF
        CASE 5: ' input TC, fCO2 or pCO2
                CALL CalculatepHfromTCfCO2(TC, fCO2, K0, K1, K2, pH)
                IF pH = -999! THEN
                        IF Batch$ = "NO" THEN CALL PrintTCfCO2Warning
                        IF Batch$ = "YES" THEN
                                IF MVFlag$ = "Y" THEN
                                        WRITE #4, -999, -999, -999, -999, -999,
-999, -999, -999, -999, -999, -999, -999, -999
                                        WRITE -999, -999, -999, -999, -999, -
999, -999, -999, -999, -999, -999, -999, -999
                                ELSE
                                        WRITE #4, -999, -999, -999, -999, -999,
-999, -999, -999, -999, -999, -999, -999
                                        WRITE -999, -999, -999, -999, -999, -
999, -999, -999, -999, -999, -999, -999
                                END IF
                                GOTO InputHereForBatchMode:
                        END IF
                        GOTO Start:
                END IF
                CALL CalculateTAfromTCpH(TC, pH, K(), T(), TA)
                IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                IF Batch$ = "YES" THEN
                        Param3 = TA * 1000000!
                        Param4 = pH
                        PRINT #4, USING " ####.#, #.####, "; Param3; Param4;
                        PRINT USING " ####.#, #.####, "; Param3; Param4;
                END IF
        CASE 6: ' input pH, fCO2 or pCO2
                CALL CalculateTCfrompHfCO2(pH, fCO2, K0, K1, K2, TC)
                CALL CalculateTAfromTCpH(TC, pH, K(), T(), TA)
                IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) =
PAlk(Dickson) + TP
                IF Batch$ = "YES" THEN
                        Param3 = TA * 1000000!
                        Param4 = TC * 1000000!
                        PRINT #4, USING " ####.#, ####.#, "; Param3; Param4;
                        PRINT USING " ####.#, ####.#, "; Param3; Param4;
                END IF
        END SELECT
        pHinp = pH: fCO2inp = fCO2: pCO2inp = pCO2
'
'
'****************************************************************************
CalculateOtherStuffAtInputConditions:
        CALL CalculateAlkParts(pH, TC, K(), T(), HCO3, CO3, BAlk, OH, PAlk,
SiAlk, Hfree, HSO4, HF)
        IF WhichKs% = 7 THEN PAlk = PAlk + T(4): ' PAlk(Peng) = PAlk(Dickson) +
TP
                HCO3inp = HCO3: CO3inp = CO3: CO2inp = TC - CO3inp - HCO3inp
                BAlkinp = BAlk: OHinp = OH: PAlkinp = PAlk: SiAlkinp = SiAlk
        CALL RevelleFactor(WhichKs%, TA, TC, K0, K(), T(), Revelle)
                Revelleinp = Revelle
        K1 = K(1): K2 = K(2)
        CALL CaSolubility(WhichKs%, Sal, TempC, Pdbar, TC, pH, K1, K2, OmegaCa,
OmegaAr)
                OmegaCainp = OmegaCa: OmegaArinp = OmegaAr
        xCO2dryinp = pCO2inp / VPFac: ' this assumes pTot = 1 atm
'
'
'****************************************************************************
CalculatepHfCO2AtOutputConditions:
        TempC = TempCout: Pdbar = Pdbarout
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
'
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatepHfromTATC(TA, TC, K(), T(), pH)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 = fCO2 /
FugFac
'
        pHout = pH: fCO2out = fCO2: pCO2out = pCO2
'
'
'****************************************************************************
CalculateOtherStuffAtOutputConditions:
        CALL CalculateAlkParts(pH, TC, K(), T(), HCO3, CO3, BAlk, OH, PAlk,
SiAlk, Hfree, HSO4, HF)
        IF WhichKs% = 7 THEN PAlk = PAlk + T(4): ' PAlk(Peng) = PAlk(Dickson) +
TP
                HCO3out = HCO3: CO3out = CO3: CO2out = TC - CO3out - HCO3out
                BAlkout = BAlk: OHout = OH: PAlkout = PAlk: SiAlkout = SiAlk
        CALL RevelleFactor(WhichKs%, TA, TC, K0, K(), T(), Revelle)
                Revelleout = Revelle
        CALL CaSolubility(WhichKs%, Sal, TempC, Pdbar, TC, pH, K1, K2, OmegaCa,
OmegaAr)
                OmegaCaout = OmegaCa: OmegaArout = OmegaAr
        xCO2dryout = pCO2out / VPFac: ' this assumes pTot = 1 atm
'
'
'****************************************************************************
        IF Batch$ = "YES" THEN
                IF fORp$ = "f" THEN fORpout = fCO2 * 1000000!
                IF fORp$ = "p" THEN fORpout = pCO2 * 1000000!
                Print4How$ = "##.####, ####.#, ##.##, ##.##, ##.##, ##.##,
####.#, ###.#, ####.#, ###.# "
                PRINT #4, USING Print4How$; pHout; fORpout; OmegaCainp;
OmegaArinp; OmegaCaout; OmegaArout; HCO3inp * 1000000!; CO3inp * 1000000!;
HCO3out * 1000000!; CO3out * 1000000!;
                PRINT USING Print4How$; pHout; fORpout; OmegaCainp; OmegaArinp;
OmegaCaout; OmegaArout; HCO3inp * 1000000!; CO3inp * 1000000!; HCO3out *
1000000!; CO3out * 1000000!;
                IF MVFlag$ = "Y" THEN
                        PRINT #4, ", "; : WRITE #4, MVFlag
                        PRINT ", "; : WRITE MVFlag
                ELSE
                        PRINT #4, ""
                        PRINT ""
                END IF
                GOTO InputHereForBatchMode:
        END IF
'
'
DoPartialsHere:
        CALL PrintHeader(ICase%, pHScale$, fORp$, TA, TC, pHinp, fCO2inp,
pCO2inp, TP, TSi, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout)
        SELECT CASE ICase%
        CASE 1: ' input TA, TC
                CALL Case1Partials(pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TA, TC, pHinp, fCO2inp,
pCO2inp, pHout, fCO2out, pCO2out)
        CASE 2: ' input TA, pH
                CALL Case2Partials(pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TA, pHinp, TC, fCO2inp,
pCO2inp, pHout, fCO2out, pCO2out)
        CASE 3: ' input TA, fCO2 or pCO2
                CALL Case3Partials(pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TA, fCO2inp, pCO2inp, TC,
pHinp, pHout, fCO2out, pCO2out)
        CASE 4: ' input TC, pH
                CALL Case4Partials(pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TC, pHinp, TA, fCO2inp,
pCO2inp, pHout, fCO2out, pCO2out)
        CASE 5: ' input TC, fCO2 or pCO2
                TCfCO2Flag% = 0
                CALL Case5Partials(pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, TC, fCO2inp, pCO2inp, TA,
pHinp, pHout, fCO2out, pCO2out, TCfCO2Flag%)
                IF TCfCO2Flag% = 1 THEN
                        TCfCO2Flag% = 0
                        CALL PrintTCfCO2Warning
                        GOTO Start:
                END IF
        CASE 6: ' input pH, fCO2 or pCO2
                CALL Case6Partials(pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal,
K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout, pHinp, fCO2inp, pCO2inp, TA,
TC, pHout, fCO2out, pCO2out)
        END SELECT
        CALL PrintKComments(WhichKs%)
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 24, 10: INPUT "Enter <A> for another sample, <E> to end, <enter>
to continue. ", Q$
        IF Q$ = "A" OR Q$ = "a" THEN GOTO Start:
        IF Q$ = "E" OR Q$ = "e" THEN END
'
'
'****************************************************************************
PrintSpeciationPage:
        CALL PrintHeader(ICase%, pHScale$, fORp$, TA, TC, pHinp, fCO2inp,
pCO2inp, TP, TSi, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout)
        CALL PrintSpeciation(WhichKs%, TA, TC, HCO3inp, CO3inp, BAlkinp, OHinp,
PAlkinp, SiAlkinp, HCO3out, CO3out, BAlkout, OHout, PAlkout, SiAlkout)
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 24, 1: INPUT "Enter <B> to go back, <A> for another sample, <E>
to end, <enter> to continue.", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO DoPartialsHere:
        IF Q$ = "A" OR Q$ = "a" THEN GOTO Start:
        IF Q$ = "E" OR Q$ = "e" THEN END
'
'
PrintfpCO2OmegasRevellePage:
        CALL PrintHeader(ICase%, pHScale$, fORp$, TA, TC, pHinp, fCO2inp,
pCO2inp, TP, TSi, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout)
        CALL PrintfpCO2OmegasRevelle(WhichKs%, fCO2inp, pCO2inp, xCO2dryinp,
Revelleinp, OmegaCainp, OmegaArinp, fCO2out, pCO2out, xCO2dryout, Revelleout,
OmegaCaout, OmegaArout)
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 24, 1: INPUT "Enter <B> to go back, <A> for another sample, <E>
to end, <enter> to continue.", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO PrintSpeciationPage:
        IF Q$ = "A" OR Q$ = "a" THEN GOTO Start:
        IF Q$ = "E" OR Q$ = "e" THEN END
'
'
PrintpHspKsPage:
        CALL PrintHeader(ICase%, pHScale$, fORp$, TA, TC, pHinp, fCO2inp,
pCO2inp, TP, TSi, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout)
        CALL PrintpHspKs(pHScale$, WhichKs%, WhoseKSO4%, TA, TC, Sal, K(), T(),
TempCinp, Pdbarinp, TempCout, Pdbarout)
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 24, 1: INPUT "Enter <B> to go back, <E> to end, <enter> to
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO PrintfpCO2OmegasRevellePage:
        IF Q$ = "E" OR Q$ = "e" THEN END
        GOTO Start:
'
'
END
'****************************************************************************
'****************************************************************************
ErrorHandler:
        BEEP: BEEP
        IError% = ERR
        CALL ErrorSub(IError%, Error$)
'
'
PrintError:
        LOCATE 20, 1
        PRINT SPACE$(80); : PRINT SPACE$(80); : PRINT SPACE$(80); : PRINT
SPACE$(80);
        LOCATE 20, 1: PRINT USING "    ERROR ###: "; ERR;
        PRINT Error$
        IF WhereFlag% = 1 AND ERR = 53 THEN PRINT "Input file "; InputFile$; "
is not found. "
        IF WhereFlag% = 2 AND ERR = 55 THEN PRINT "Output file "; OutputFile$; "
is same as input file. "
        IF WhereFlag% = 3 AND Batch$ = "YES" THEN PRINT NRead% - 1; "lines have
been input. "
        IF ERR = 62 THEN
                PRINT "This may be due to blank lines at the end of ";
InputFile$
                PRINT "    or the wrong number of header lines listed for ";
InputFile$
        END IF
'
'
DecideWhatToDo:
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "Choose <enter> to restart, <E> to end, or <C> to continue: ", Q$
        SELECT CASE Q$
                CASE ""
                        CLOSE
                        RESUME Start:
                CASE "E", "e"
                        CLOSE
                        END
                CASE "C", "c"
                        RESUME
                CASE ELSE
                        GOTO DecideWhatToDo:
        END SELECT
'****************************************************************************
' the lines below are for possible inclusion later.
'PRINT "    THE LAST ID ENTERED WAS "; ID$
'INPUT "    ENTER C TO CONTINUE WITH THE NEXT VALUE ", ANS$
'PRINT #4, "THERE WAS AN ERROR, THE LAST ID ENTERED IS "; ID$
'RESUME InputData:
SUB AboutCO2SYS (Info$)
' SUB AboutCO2SYS, version 03.04, 10-15-97, written by Ernie Lewis.
' Inputs: Info$
' Outputs: none
' This prints information about the program CO2SYS.
'
'
TopOfInfoSub:
        SELECT CASE Info$
                CASE ""
                        EXIT SUB
                CASE "General"
                        GOSUB Generalp1:
                CASE "Menu"
                        GOSUB PrintInfoMenu:
                CASE "pHScales"
```

```
                        GOSUB AboutpHScalesp1:
                CASE "fCO2pCO2"
                        GOSUB AboutfCO2pCO2:
                CASE "KSO4"
                        GOSUB AboutKSO4:
                CASE "Batch"
                        GOSUB AboutBatchModep1:
                CASE "Constants"
                        GOSUB AboutConstantsp1:
                CASE "GEOSECS"
                        GOSUB AboutGEOSECSp1:
                CASE "Peng"
                        GOSUB AboutPeng:
                CASE "Freshwater"
                        GOSUB AboutFreshwaterOption:
                CASE "Pressure"
                        GOSUB AboutPressureEffects:
                CASE "Calcium"
                        GOSUB AboutCalciumSolubility:
                CASE "Alkalinity"
                        GOSUB AboutAlkalinity:
                CASE "Revelle"
                        GOSUB AboutRevelleFactor:
        END SELECT
        GOTO TopOfInfoSub:
EXIT SUB
'****************************************************************************
PrintInfoMenu:
        CLS
        PRINT "   For more information, choose one of the following: "
        PRINT
        PRINT "         1) General information "
        PRINT "         2) pH scales "
        PRINT "         3) fCO2, pCO2 "
        PRINT "         4) KSO4 "
        PRINT "         5) Batch-input mode "
        PRINT "         6) GEOSECS option "
        PRINT "         7) Peng option "
        PRINT "         8) Freshwater option "
        PRINT "         9) Pressure effects "
        PRINT "         10) Calcium carbonate solubility (omega values) "
        PRINT "         11) Alkalinity "
        PRINT "         12) Revelle factor "
        PRINT "         13) Constants used "
        PRINT
        PRINT "    Enter <E> to end, <enter> to continue. "
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 55: INPUT "", Q$
        IF Q$ = "E" OR Q$ = "e" THEN CLS : END
        Q% = VAL(Q$)
        SELECT CASE Q%
                CASE 1
                        Info$ = "General"
                CASE 2
                        Info$ = "pHScales"
                CASE 3
```

```
                        Info$ = "fCO2pCO2"
                CASE 4
                        Info$ = "KSO4"
                CASE 5
                        Info$ = "Batch"
                CASE 6
                        Info$ = "GEOSECS"
                CASE 7
                        Info$ = "Peng"
                CASE 8
                        Info$ = "Freshwater"
                CASE 9
                        Info$ = "Pressure"
                CASE 10
                        Info$ = "Calcium"
                CASE 11
                        Info$ = "Alkalinity"
                CASE 12
                        Info$ = "Revelle"
                CASE 13
                        Info$ = "Constants"
                CASE ELSE
                        Info$ = ""
        END SELECT
        GOTO TopOfInfoSub:
RETURN
'****************************************************************************
Generalp1:
        CLS
        PRINT "    Program CO2SYS, version 01.05, written by Ernie Lewis. "
        PRINT
        PRINT "    This program takes two parameters of the CO2 system in
seawater (TA, TC, "
        PRINT "    pH, fCO2 or pCO2), and calculates the other two at a set of
input "
        PRINT "    conditions (T and P) and a set of output conditions chosen by
the user. "
        PRINT "    It supersedes the 1995 programs CO2SYSTM, FCO2TCO2, PHTCO2,
and CO2BTCH. "
        PRINT
        PRINT "    For questions, comments, or to report any problems, please
contact: "
        PRINT "         Ernie Lewis or Doug Wallace "
        PRINT "         Department of Applied Science "
        PRINT "         Building 318 "
        PRINT "         P. O. Box 5000 "
        PRINT "         Brookhaven National Laboratory "
        PRINT "         Upton, NY 11973-5000 "
        PRINT
        PRINT "         elewis@bnl.gov     wallace@bnl.gov "
        PRINT "         516-344-7406       516-344-2945 "
        PRINT
        PRINT "    This work was supported by the US Department of Energy Office
of Health and "
        PRINT "    Enviromental Research under contract DE-ACO2-76CH00016,
through a project "
```

```
        PRINT "    entitled `Inorganic Carbon for the World Ocean Circulation
Experiment - "
        PRINT "    World Hydrographic Program' (D.W.R. Wallace and K.M. Johnson,
PIs). "
        PRINT
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 24, 1
        INPUT "    Enter <P> to go to the program, <enter> to continue. ", Q$
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
Generalp2:
        CLS
        PRINT "    Every effort has been made to make this program as correct,
complete, and "
        PRINT "    user-friendly as possible. HOWEVER, the program is not
failsafe and some "
        PRINT "    familiarity with the CO2 system in seawater is assumed. "
        PRINT
        PRINT "    The effects of phosphate, silicate, and OH are included, as
well as the "
        PRINT "    non-ideality of CO2. Some programs we have evaluated do not
include these, "
        PRINT "    which can have a significant effect on the results. "
        PRINT
        PRINT "    In developing this program, much work was done to ensure that
correct "
        PRINT "    values for the various constants were used. There is a
paucity of data "
        PRINT "    for many of the values. Many errors were found in the
literature. Whenever "
        PRINT "    possible these were corrected or otherwise noted. A listing
is included "
        PRINT "    in the accompanying documentation. "
        PRINT
        PRINT "    This program is designed for either single-input mode or
batch-input mode, "
        PRINT "    and allows for a variety of options, including: "
        PRINT "         choice of various formulations for K1 and K2, "
        PRINT "         two distinct formulations for KSO4 (Dickson's or
Khoo's), "
        PRINT "         choice of four pH scales (free, total, seawater, or
NBS), "
        PRINT "         use of either fugacity (fCO2) or partial pressure (pCO2)
of CO2, "
        PRINT "         and choice of any two CO2 system parameters as inputs. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <P> to go to the program, <enter> to
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO Generalp1:
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
Generalp3:
        CLS
        PRINT "    In single-input mode, after selection of the various options
for the value "
        PRINT "    of the constants, etc. the user is prompted for the
following: "
        PRINT "         the salinity, "
        PRINT "         the concentrations of silicate and phosphate, "
        PRINT "         the input temperature and pressure (or depth), "
        PRINT "         the output temperature and pressure (or depth), "
        PRINT "         the two known CO2 system parameters at the input
conditions. "
        PRINT "    Default values, given in parentheses, will be used when
<enter> is input. "
        PRINT
        PRINT "    The program will calculate the other two CO2 system
parameters at the input "
        PRINT "    conditions. TA and TC, which do not vary with temperature and
pressure, "
        PRINT "    are used to calculate the pH and fCO2 (or pCO2) at the output
conditions. "
        PRINT
        PRINT "    Also calculated for both the input and the output conditions
are: "
        PRINT "         sensitivity of the output results to small variations of
the inputs, "
        PRINT "         contributions to the alkalinity and carbon speciation,
"
        PRINT "         fCO2 and pCO2, "
        PRINT "         omega (the degree of saturation) for calcite and for
aragonite, "
        PRINT "         the Revelle, or homogeneous buffer, factor, "
        PRINT "         pH values on all four pH scales, "
        PRINT "         the values of pK1, pK2, pKW, and pKB. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <P> to go to the program, <enter> to
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO Generalp2:
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
        Info$ = "Menu"
RETURN
'****************************************************************************
AboutpHScalesp1:
        CLS
        PRINT "    The various pH scales are inter-related by the following
equations: "
        PRINT
        PRINT "           -pHNBS                            Htot
Hsws        "
        PRINT "    aH = 10      = fH * Hsws,  Hfree  =  -----------  =
------------------ "
        PRINT "                                         1 + TS/KSO4     1 +
TS/KSO4 + TF/KF "
        PRINT
        PRINT "    where aH is the activity and fH the activity coefficient of
the H+ ion "
```

```
        PRINT "    (this includes liquid junction effects), TS and TF are the
concentrations "
        PRINT "    of SO4- and fluorine, and KSO4 and KF are the dissociation
constants of "
        PRINT "    HSO4 and HF in seawater. "
        '(which are inherently on the free scale)
        PRINT
        PRINT "    These conversions depend on temperature, salinity, and
pressure. "
        PRINT "    At 20 deg C, Sal 35, and 1 atm, pH values on the total scale
are (about) "
        PRINT "         .09 units lower than those on the free scale, "
        PRINT "         .01 units higher than those on the seawater scale, and "
        PRINT "         .13 units lower than those on the NBS scale. "
        PRINT
        PRINT "    The concentration units for aH on the NBS scale are mol/kg-
H2O. "
        PRINT "    The concentration used here for [H] on the other scales is
mol/kg-SW "
        PRINT "    (note that the free scale was originally defined in units of
mol/kg-H2O). "
        PRINT "    The difference between mol/kg-SW and mol/kg-H2O is about .015
pH units "
        PRINT "    at salinity 35 (the difference is nearly proportional to
salinity). "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <P> to go to the program, <enter> to continue. ", Q$
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
AboutpHScalesp2:
        CLS
        PRINT "    The seawater scale was formerly referred to as the total
scale, and "
        PRINT "    each is still sometimes referred to as the other in the
literature. "
        PRINT "    The fit of fH used here is valid from salinities 20 to 40. "
        PRINT "    fH has been found to be electrode-dependent, and does NOT
equal 1 at "
        PRINT "    salinity 0 due to the liquid junction potential. "
        PRINT "    Values on the NBS pH scale are only accurate to (at
best) .005. "
        PRINT "    All work on pressure effects on pH has assumed that fH is
independent "
        PRINT "    of pressure. Some of the pH scale conversions depend on
pressure. "
        PRINT
        PRINT "    For discussions of the various pH scales, see: "
        PRINT "    Dickson, Deep-Sea Research 40:107-118, 1993, "
        PRINT "    Millero, Marine Chemistry 44:143-152, 1993, "
        PRINT "    Dickson, Geochemica et Cosmochemica Acta 48:2299-2308, 1984,
"
        PRINT "    Butler, Marine Chemistry 38:251-282, 1992, "
        PRINT "    Culberson, C. H., Direct Potentiometry, Chapter 6 (pp. 187-
261), in: "
```

```
        PRINT "         Marine Electrochemistry, eds. M. Whitfield and D.
Jagner, 1981. "
        PRINT
        PRINT "    Attention is required because in some of these the
distinction between the "
        PRINT "    total and seawater pH scales was not made. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <enter> to continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutpHScalesp1:
        Info$ = "Menu"
RETURN
'*****************************************************************************
AboutfCO2pCO2:
        CLS
        PRINT "    The fugacity of CO2 (fCO2) in water is defined to be the
fugacity of CO2 "
        PRINT "    in wet (100% water-saturated) air which is in equilibrium
with the water. "
        PRINT "    pCO2, the partial pressure of CO2, is defined to be the
product of the "
        PRINT "    mole fraction of CO2 in WET air and the total pressure. This
is the "
        PRINT "    same as the product of the the mole fraction of CO2 in DRY
air (xCO2(dry)) "
        PRINT "    and (pTot - pH2O), where pH2O is the vapor pressure of water
above seawater."
        PRINT "    At pressures of order 1 atm fCO2 in air is about .3% lower
than the pCO2 due"
        PRINT "    to the non-ideality of CO2 (see Weiss, Marine Chemistry
2:203-215, 1974). "
        PRINT "    This program assumes a pressure near 1 atm (where most
equilibrators "
        PRINT "    function) for the conversion between partial pressure and
fugacity. "
        PRINT
        PRINT "    fCO2 is related to TC and pH by the following equation: "
        PRINT "                 [CO2*]     TC          H*H          "
        PRINT "         fCO2 =  ------  =  -- *  ------------------ "
        PRINT "                   K0       K0    H*H + K1*H + K1*K2 "
        PRINT "    where [CO2*] is the concentration of dissolved CO2, K0 is the
solubility "
        PRINT "    coefficient of CO2 in seawater, and K1 and K2 are the first
and second "
        PRINT "    dissociation constants for carbonic acid in seawater. "
        PRINT
        PRINT "    Units for fCO2 and pCO2 in this program are uatm (micro-
atmospheres). "
        PRINT "    The value of xCO2(dry) given in this program assumes pTot = 1
atmosphere. "
        PRINT "    GEOSECS and Peng et al did not distinguish between fCO2 and
pCO2, nor did "
        PRINT "    some other programs that we have evaluated. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
        '''''''''''''''''''''''''''''''''''''
        ' pCO2(wet) = xCO2(wet) * pTot = xCO2(dry) * (pTot - VPSW)
        ' where VPSW is the vapor pressure of water above seawater
        ' fCO2(wet) = pCO2(wet) * FugFac
        ' (pTot - VPSW) converts from wet air to dry air
        ' FugFac converts partial pressure to fugacity
        '''''''''''''''''''''''''''''''''''''
RETURN
'****************************************************************************
AboutKSO4:
        CLS
        PRINT "    KSO4 is defined to be the dissociation constant for the 
reaction "
        PRINT "         HSO4- = H+ + SO4--, "
        PRINT "    thus KSO4 = [H] * [SO4] / [HSO4]. "
        PRINT
        PRINT "    Two formulations of this are still in current usage: "
        PRINT "    Khoo et al, Analytical Chemistry, 49(1):29-34, 1977, and "
        PRINT "    Dickson, Journal of Chemical Thermodynamics, 22:113-127, 
1990. "
        PRINT
        PRINT "    The values of Dickson are now recommended, though many older 
papers used "
        PRINT "    values of Khoo et al. They are between 15 to 45 % lower than 
those of "
        PRINT "    Dickson, depending on temperature (mostly). "
        PRINT
        PRINT "    The main effect of this difference will occur when converting 
from one "
        PRINT "    pH scale to another, or when working on a scale for which 
equilibrium "
        PRINT "    constants must be converted (e.g., most constants were 
determined on "
        PRINT "    either the total scale or the seawater scale). "
        PRINT
        PRINT "    Use of the Dickson values when converting from the total pH 
scale to the "
        PRINT "    free pH scale will result in pH values which are .015 to .03 
units lower "
        PRINT "    than those obtained using values of Khoo et al. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
RETURN
'****************************************************************************
AboutBatchModep1:
        CLS
        PRINT "    Batch-input mode is designed to be used with large data sets 
such as "
        PRINT "    files created by EXCEL or other spreadsheet programs. Data 
are read from "
        PRINT "    an input file, and results are calculated and printed to an 
output file. "
        PRINT
        PRINT "    After the various options are chosen, the user is prompted 
for: "
```

```
        PRINT "         the name of the input file, "
        PRINT "         the name of the output file, "
        PRINT "         the number of header lines in the input file, "
        PRINT "         the number of ID fields preceding the data for each
sample, "
        PRINT "         a numeric value to denote missing data, and "
        PRINT "         whether or not to flag samples containing missing data.
"
        PRINT
        PRINT "    Each input line must contain the following fields for one
sample: "
        PRINT "         the ID fields (the number of which is selected by the
user), "
        PRINT "         salinity (on the practical salinity scale), "
        PRINT "         total phosphate (in umol/kg-SW), "
        PRINT "         total silicate (in umol/kg-SW), "
        PRINT "         temperature (deg C) for input conditions, "
        PRINT "         pressure (dbar) for input conditions, "
        PRINT "         temperature (deg C) for output conditions, "
        PRINT "         pressure (dbar) for output conditions, "
        PRINT "         the first known CO2 system parameter, and "
        PRINT "         the other known CO2 system parameter. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <P> to go to the program, <enter> to continue. ", Q$
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
AboutBatchModep2:
        CLS
        PRINT "    It is VERY IMPORTANT that the input data are in the correct
format and "
        PRINT "    that the correct order of the CO2 system parameters is
followed: "
        PRINT "         TA will always come first if used; "
        PRINT "         TC (if used) will precede any parameter except TA; "
        PRINT "         pH (if used) is next; and "
        PRINT "         fCO2 (or pCO2) will always be last if used. "
        PRINT "    Units used are umol/kg-SW and uatm. "
        PRINT
        PRINT "    Six example data files: CASE1.INP, CASE2.INP, etc. are
included, one "
        PRINT "    for each of the choices of CO2 system input parameters. "
        PRINT
        PRINT "    The input data may be space-separated or comma-separated
(recommended). "
        PRINT "    If the data are space-separated data the ID fields MUST be
within double "
        PRINT "    quotes, but for comma-separated data this is not required.
(Note that EXCEL "
        PRINT "    puts double quotes around each double quote when importing a
CSV file). "
        PRINT
        PRINT "    It can occur that for certain inputs of TC and fCO2 (or pCO2)
the system "
```

```
        PRINT "    has no solution for the given input conditions. If this
occurs, -999 is "
        PRINT "    printed for TA, pH, and the other calculated parameters. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <P> to go to the program, <enter> to
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutBatchModep1:
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
AboutBatchModep3:
        CLS
        PRINT "    Since a data set may contain values which are missing or
unknown, the user "
        PRINT "    may define a numeric value to be the missing-value designator
(MVD). If one "
        PRINT "    is not defined by the user, the default value -9 is used. IT
IS IMPERATIVE "
        PRINT "    THAT A VALUE IS PROVIDED IN EACH OF THE INPUT DATA FIELDS, so
unknown "
        PRINT "    values should always be set equal to the MVD. "
        PRINT
        PRINT "    If the MVD is input for one of the CO2 system parameters, no
calculations "
        PRINT "    are made and each output variable is given the value of the
MVD. If the "
        PRINT "    MVD is input for one of the non-CO2 system parameters, a
default value is "
        PRINT "    used in calculations and that sample is flagged (if that
option has been "
        PRINT "    chosen). HOWEVER, the MVD will be printed in the output file
for that "
        PRINT "    variable. OBVIOUSLY, CAUTION SHOULD BE USED IN INTERPRETING
THE RESULTS "
        PRINT "    WHEN THERE ARE MISSING VALUES IN THE INPUT FILE. "
        PRINT
        PRINT "     The defaults used are: "
        PRINT "         salinity: 35 "
        PRINT "         total phosphate: 0 umol/kg-SW "
        PRINT "         total silicate: 0 umol/kg-SW "
        PRINT "         input temperature: 20 deg "
        PRINT "         input pressure: 0 dbar "
        PRINT "         output temperature: the value of the input temperature "
        PRINT "         output pressure: the value of the input pressure. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <P> to go to the program, <enter> to
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutBatchModep2:
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
AboutBatchModep4:
        CLS
        PRINT "    The output file will contain header lines with the following:
"
        PRINT "         the input file name and the date, "
        PRINT "         the choice of constants, KSO4, and pH scale, "
        PRINT "         any other header lines input by the user, and "
        PRINT "         labels for the output fields. "
        PRINT
        PRINT "    The output data are comma-separated with one line per sample
containing the "
        PRINT "    same fields as the input data plus the following calculated
values: "
        PRINT "         the other two CO2 system parameters at the input
conditions, "
        PRINT "         pH and fCO2 (or pCO2) at the output conditions, "
        PRINT "         omegas for calcite and aragonite at the input
conditions, "
        PRINT "         [HCO3-] and [CO3--] at the input conditions, "
        PRINT "         omegas for calcite and aragonite at the output
conditions, and "
        PRINT "         [HCO3-] and [CO3--] at the output conditions. "
        PRINT "    If the user chose to flag missing data, an extra field is
appended "
        PRINT "    with the value MVD if there were missing data, 0 otherwise. "
        PRINT
        PRINT "    The pH values are reported on the scale chosen by the user. "
        PRINT
        PRINT "    To load the output file into EXCEL, simply open it as comma-
separated "
        PRINT "    (CSV). "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <enter> to continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutBatchModep3:
        Info$ = "Menu"
RETURN
'****************************************************************************
AboutFreshwaterOption:
        CLS
        PRINT "    For the freshwater option only [HCO3], [CO3], [OH], and [H]
are included "
        PRINT "    in the definition of alkalinity: TA = [HCO3] + 2[CO3] + [OH]
- [H]. "
        PRINT
        PRINT "    fH, the activity coefficient of H+, does NOT equal 1 at
salinity 0 due "
        PRINT "    to liquid junction effects (included in its definition). It
is also "
        PRINT "    found to be electrode dependent. Thus, while the values of pH
on the "
        PRINT "    free, total, and seawater scales will coincide at salinity 0,
the value "
        PRINT "    on the NBS scale will differ. For these reasons, for this
choice only a "
        PRINT "    pH value is given without reference to a pH scale. "
        PRINT
'        PRINT "    Only one set of measurements of K1 and K2 has been made in
seawater at "
```

```
 '       PRINT "    salinity < 10. Though the values can be extrapolated to
salinity 0 they "
  '      PRINT "    change by a considerable amount over this interval (between
salinities 0 "
   '     PRINT "    and 5, K1 varies by a factor of 2 and K2 by between 6.5 and
9.2, depending "
    '    PRINT "    on temperature). For comparison, between salinities 5 and 35
K1 varies by "
     '   PRINT "    a factor of less than 1.5 and K2 less than 3). Thus a fit of
K1 and K2 for "
      '  PRINT "    values of salinity in this range would be prone to large
uncertainty. For "
       ' PRINT "    this reason, only values of K1 and K2 valid at salinity 0
are used."
        'PRINT
        PRINT "    Constants used for this choice (K1, K2, and KW) are from: "
        PRINT "         Millero, F. J., Geochemica et Cosmochemica Acta 43:1651-
1661, 1979. "
        PRINT "    Pressure effects on these constants are from: "
        PRINT "         Millero, Chap. 43, Chemical Oceanography, ed. Riley +
Chester, 1983. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
        ''''''''''''''''''''''''''''''
        'PRINT "    Further, it is inherent in the determination of K1 and K2
that the "
        'PRINT "         seawater ratios of constituents occur, which is
unlikely, "
        'PRINT "         making it hard to define what is meant by salinity. "
        ' at 20 deg, K1(5) / K1(0) = 2.0, K1(35) / K1(5) = 1.4
        '            K2(5) / K2(0) = 8.5, K2(35) / K2(5) = 2.8
        ''''''''''''''''''''''''''''''
RETURN
'****************************************************************************
AboutGEOSECSp1:
        CLS
        PRINT "    The GEOSECS option was designed to replicate the calculations
performed "
        PRINT "    in Chapter 3, Carbonate Chemistry, by Takahashi et al, in
GEOSECS Pacific "
        PRINT "    Expedition, Volume 3, by Broecker et al, 1982. "
        PRINT
        PRINT "    That work used the NBS pH scale, the values of K1 and K2 from
Mehrbach "
        PRINT "    et al, and the value of KB from Lyman. It did not include
effects of OH, "
        PRINT "    silicate, or phosphate, nor was there a correction for the
non-ideality "
        PRINT "    of CO2 (i.e., implying fCO2 and pCO2 are the same). Their
boron "
        PRINT "    concentration was about 1% lower than that used for the other
choices in "
        PRINT "    this program (except the choice of Peng). "
        PRINT
        PRINT "    In GEOSECS, TA and TC values from titration were used to
determine pCO2, "
        PRINT "    [H2CO3], [HCO3-], [CO3--], and pH, at P = 1 atm and insitu T;
and "
        PRINT "    [H2CO3], [HCO3-], [CO3--], aH, pH, ICP, and delta CO3-- for
calcite and "
        PRINT "    aragonite at insitu T and P, where aH = 10^(-pH), ICP = [Ca+
+][CO3--], and "
        PRINT "    delta CO3-- is the difference between [CO3--] and its
saturation level. "
        PRINT
        PRINT "    These last three parameters were used to describe the
saturation states of "
        PRINT "    calcite and aragonite. In this program only omegas,
dimensionless ratios, "
        PRINT "    are output for this. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <P> to go to the program, <enter> to continue. ", Q$
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'****************************************
AboutGEOSECSp2:
        CLS
        PRINT "    A fit for fH was also given (for salinities 20 to 40) and is
used to "
        PRINT "    convert between pH scales in this program. "
        PRINT
        PRINT "    Some typographic errors in the GEOSECS report were noted and
corrected: "
        PRINT "         in the pressure dependence of K2 the given value 26.4
should be 16.4, "
        PRINT "         and the expression for ln KW should have C*ln T, not
C/ln T. "
        PRINT "    That these are correct can be seen by checking the original
references. "
        PRINT
        PRINT "    The ratio of Ksp(arag.) / Ksp(calc.) is given as 1.48 in the
original "
        PRINT "    reference (Berner, R. A., American Journal of Science
276:713-730, 1976), "
        PRINT "    but the value of 1.45 given in GEOSECS was used both in that
work and in "
        PRINT "    this program as well for this choice. "
        PRINT
        PRINT "    The GEOSECS report also contains a discussion on the effects
of OH, "
        PRINT "    phosphate, and silicate (see pp. 79-82, especially Table 1 on
p. 81, of "
        PRINT "    Chapter 3, Carbonate Chemistry, by Takahashi et al, in
GEOSECS Pacific "
        PRINT "    Expedition, V. 3, by Broecker et al, 1982). From this, it can
be seen how "
        PRINT "    important these can be, especially for calculated values of
fCO2 (or pCO2). "
```

```
        PRINT "    This table has a typo: 17.8 for Aw in Pacific Surface Water 
should be 7.8. "
        PRINT
        PRINT "    The choice of Peng is very similar, and should be used 
instead if the "
        PRINT "    values of OH, etc. are desired with these constants. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <enter> to continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutGEOSECSp1:
        Info$ = "Menu"
RETURN
'***************************************************************************
AboutPeng:
        CLS
        PRINT "    This choice replicates the calculation scheme of Peng et al, 
Tellus 39B: "
        PRINT "    439-458, 1987, which is similar to GEOSECS. Peng et al worked 
on the NBS "
        PRINT "    pH scale and included effects of phosphate, silicate, and OH, 
but did not "
        PRINT "    distinguish between fCO2 and pCO2. The values of K1 and K2 
from Mehrbach et "
        PRINT "    al and the value of KB from Lyman were used. "
        PRINT
        PRINT "    They did not treat calcite and aragonite solubility or 
pressure effects, "
        PRINT "    but these are included in this program for this choice using 
GEOSECS values "
        PRINT "    for solubility and pressure dependence of K1, K2, and KB, and 
the same "
        PRINT "    values for the pressure dependence of OH and phosphate and 
silicate "
        PRINT "    dissociation as are used in constant choices 1 to 5. The 
concentration "
        PRINT "    of boron they used was about 1% lower than that used for 
other choices in "
        PRINT "    this program (except for GEOSECS choice). "
        PRINT
        PRINT "    The value of fH given in their paper was NOT the same as that 
given in the "
        PRINT "    GEOSECS report as claimed, rather it had been rounded off and 
was therefore "
        PRINT "    about 1% higher, corresponding to a change of .003 in pH. 
Note that the "
        PRINT "    check value given in the paper does not match either fit. "
        PRINT
        PRINT "    Their definition of alkalinity (TA) differs from that of 
Dickson (Deep-Sea "
        PRINT "    Research 28A:609-623, 1981 - used in constant choices 1-5 in 
this program) "
        PRINT "    in that it is greater by an amount equal to the total 
phosphate (TP). This "
        PRINT "    seems insignificant, but can affect the calculated fCO2 
appreciably. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
RETURN
'***************************************************************************
AboutPressureEffects:
        CLS
        PRINT "    The equilibrium constants depend on pressure as well as
temperature and "
        PRINT "    salinity. Data are scarce on these effects in seawater and
most values are "
        PRINT "    estimated from molal volume data. Few measurements have been
made for K1, "
        PRINT "    K2, and KB, at only a few combinations of temperature,
salinity, and "
        PRINT "    pressure (mostly in artificial seawater). All of the work
assumed that fH, "
        PRINT "    the activity coefficient of H+ (including liquid junction
effects), is "
        PRINT "    independent of pressure. Some of the pH scale conversions do
depend on "
'        PRINT "    pressure. Values of the constants should be converted to the
seawater or "
'        PRINT "    NBS pH scale WITHOUT pressure-corrected pH scale
conversions, then "
'        PRINT "    corrected for pressure, then converted back to the desired
pH scale WITH "
'        PRINT "    pressure-corrected pH scale conversions. Measurements have
also been made "
'        PRINT "    on the calcite and aragonite solubilities in seawater at
pressure. "
        PRINT "    pressure. Measurements have also been made on the calcite and
aragonite "
        PRINT "    solubilities in seawater at pressure. "
        PRINT
        PRINT "    Depth in meters and pressure in decibars are used
interchangeably in this "
        PRINT "    program. They differ by only 3% at 10000 dbar (less at lower
pressures), "
        PRINT "    well within the uncertainties of the pressure effects on the
constants. "
        PRINT "    No salinity dependence of the pressure corrections is used in
this program. "
        PRINT "    The values used are taken from: "
        PRINT "         Millero, GCA 59:661-671, 1995, table 9 on p. 675, "
        PRINT "         Millero, GCA 43:1651-1661, 1979, table 5 on p. 1657, "
        PRINT "         Millero, Chap. 43, Chemical Oceanography, ed. Riley +
Chester, 1983, "
        PRINT "    Note that some typos and inconsistencies from these papers
were corrected. "
        PRINT "         Takahashi et al, Chap. 3 in GEOSECS Pacific Expedition,
v. 3, 1982. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
RETURN
'***************************************************************************
```

```
AboutCalciumSolubility:
        CLS
        PRINT "    The solubility product (Ksp) is calculated for both calcite
and aragonite "
        PRINT "    and the saturations states are given in terms of Omega, the
solubility "
        PRINT "    ratio, defined as Omega =  [CO3--]*[Ca++] / Ksp. Thus, values
of Omega < 1 "
        PRINT "    represent conditions of undersaturation, and values of Omega
> 1 represent "
        PRINT "    conditions of oversaturation. "
        PRINT
        PRINT "    The concentration of calcium, [Ca++], is assumed to be
proportional to the "
        PRINT "    salinity, and the carbonate, [CO3--], is calculated from TC,
pH, and the "
        PRINT "    values of K1 and K2 for carbonic acid. "
        PRINT
        PRINT "    The values used in this program are from: "
        PRINT "         Mucci, American Journal of Science 283:781-799, 1983, "
        PRINT "         Ingle, Marine Chemistry 3:301-319, 1975, "
        PRINT "         Millero, Geochemica et Cosmochemica Acta 43:1651-1661,
1979, "
        PRINT "         Takahashi et al, Chap. 3, GEOSECS Pacific Expedition, v.
3, 1982, "
        PRINT "         Berner, American Journal of Science 276:713-730, 1976. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
RETURN
'****************************************************************************
AboutAlkalinity:
        CLS
        PRINT "    The definition of alkalinity (TA) used in this program for
constant choices "
        PRINT "    1 to 5 is the same as that of Dickson, Deep-Sea Research
28A:609-623, 1981: "
        PRINT "         TA = [HCO3] + 2[CO3] + [B(OH)4] + [OH] + [HPO4] + 2[PO4]
+ [SiO(OH)3] "
        PRINT "            + [HS] + 2[S] + [NH3] - [H] - [HSO4] - [HF] -
[H3PO4], "
        PRINT "    except that the contributions of HS, S, and NH3 are not
included. "
        PRINT
        PRINT "    For the choice of Peng, the definition of Peng et al, Tellus
39B:439-458, "
        PRINT "    1987 is used. The main difference is that it is greater by an
amount equal "
        PRINT "    to the total phosphate: "
        PRINT "         TP = [PO4---] + [HPO4--] + [H2PO4-] + [H3PO4]. "
        PRINT "    Though this seems small, it can have a large effect on the
calculated fCO2. "
        PRINT "    Each umol/kg-SW of TA results in a change in about .5% in
fCO2, so a value "
        PRINT "    of TP = 3 umol/kg-SW (a modest amount) can result in a
difference of "
```

```
        PRINT "    5 to 20 uatm (or more) in fCO2 between the two definitions. "
        PRINT
        PRINT "    The definition used for the GEOSECS choice is: "
        PRINT "         TA = [HCO3] + 2[CO3] + [H2BO3], "
        PRINT "    and for the freshwater choice is: "
        PRINT "         TA = [HCO3] + 2[CO3] + [OH] - [H]. "
        PRINT
        PRINT "    In this program values of alkalinity are given in micro-moles
per kilogram "
        PRINT "    of seawater (umol/kg-SW). "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
RETURN
'****************************************************************************
AboutRevelleFactor:
        CLS
        PRINT "    The Revelle, or homogeneous buffer, factor is the % change in
fCO2 "
        PRINT "    (or pCO2) caused by a 1% change in TC at constant alkalinity.
"
        PRINT
        PRINT "    It depends on temperature, salinity, and the total alkalinity
and TC "
        PRINT "    (or any combination of the two CO2 system parameters) of the
sample. "
        PRINT
        PRINT "    It is calculated at both the input and output conditions
using: "
        PRINT
        PRINT "         Revelle factor = (dfCO2/dTC) / (fCO2/TC) at constant TA.
"
        PRINT
        PRINT "    Normal seawater values are between 8 and 20. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Hit <enter> to continue. ", Q$
        Info$ = "Menu"
RETURN
'****************************************************************************
AboutConstantsp1:
        CLS
        PRINT "    Constants are converted to the appropriate pH scale and
concentration "
        PRINT "         scale, if needed, before calculations are made. "
        PRINT
        PRINT "    The value of K0 (the solubility coefficient of CO2) and the
conversion "
        PRINT "    between the fugacity and the partial pressure of CO2 are from
"
        PRINT "         Weiss, R. F., Marine Chemistry 2:203-215, 1974. "
        PRINT
        PRINT "    The vapor pressure of H2O above seawater is from "
        PRINT "         Weiss, R. F., and Price, B. A., Marine Chemistry 8:347-
359, 1980. "
        PRINT
        PRINT "    The concentrations of sulfate and fluorine are from
(respectively) "
        PRINT "         Morris and Riley, Deep-Sea Research 13:699-705, 1966,
and "
        PRINT "         Riley, J. P., Deep-Sea Research 12:219-220, 1965. "
        PRINT
        PRINT "    The value of KSO4, the dissociation constant for HSO4, is
from either "
        PRINT "         Khoo, et al, Analytical Chemistry, 49(1):29-34, 1977, or
"
        PRINT "         Dickson, Journal of Chemical Thermodynamics, 22:113-127,
1990. "
        PRINT
        PRINT "    KF, the dissociation constant for HF, is from "
        PRINT "         Dickson, A. G. and Riley, J. P., Marine Chemistry 7:89-
99, 1979. "
        PRINT
        PRINT "    Constants for calcium solubility and for pressure effects are
given in "
        PRINT "         other information sections. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <P> to go to the program, <enter> to continue. ", Q$
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'**************************************
AboutConstantsp2:
        CLS
        PRINT "    The value of KB (for boric acid), in constant choices 1 to 5,
is from "
        PRINT "         Dickson, Andrew G., Deep-Sea Research 37:755-766, 1990.
"
        PRINT "    GEOSECS and Peng choices use Lyman's KB, the fit being from "
        PRINT "         Li et al, Journal of Geophysical Research 74:5507-5525,
1969. "
        PRINT
        PRINT "    The boron concentration in constant choices 1 to 5 is from "
        PRINT "         Uppstrom, Leif, Deep-Sea Research 21:161-162, 1974. "
        PRINT "    For the GEOSECS and Peng choices, it is from "
        PRINT "         Culkin, F., in Chemical Oceanography, ed. Riley and
Skirrow, 1965. "
        PRINT
        PRINT "    Values of KW (for H2O), KP1, KP2, and KP3 (for phosphoric
acid), and "
        PRINT "         and KSi (for silicic acid) are from (in constant choices
1 to 5) "
        PRINT "         Millero, Frank J., Geochemica et Cosmochemica Acta
59:661-677, 1995 "
        PRINT "         (some typos and inconsistencies from this paper were
corrected). "
        PRINT "    The Peng choice uses KP2 and KP3 from "
        PRINT "         Kester and Pytkowicz, Limnology and Oceanography 12:243-
252, 1967, "
        PRINT "    and KSi from "
        PRINT "         Sillen, Martell, and Bjerrum, Stability constants of
metal-ion "
```

```
        PRINT "         complexes, The Chemical Society (London), Special Publ. 
17:751, 1964. "
        PRINT "    For the Peng and the freshwater choice, KW is from "
        PRINT "         Millero, F. J., Geochemica et Cosmochemica Acta 43:1651-
1661, 1979. "
        PRINT "         For the freshwater choice, the fit is a refit of data 
from "
        PRINT "         Harned and Owen, Physical Chemistry of Electrolyte 
Solutions, 1958. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <P> to go to the program, <enter> to 
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutConstantsp1:
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
'***************************************
AboutConstantsp3:
        CLS
        PRINT "    Several determinations of K1 and K2 of carbonic acid have 
been made: "
        PRINT "         Hansson (1973) on the total pH scale, "
        PRINT "         Mehrbach et al (1973) on the NBS pH scale, "
        PRINT "         Goyet and Poisson (1989) on the seawater scale, and "
        PRINT "         Roy et al (1993) on the total scale. "
        PRINT "    The data of Hansson and Mehrbach et al, both seperately and 
together, "
        PRINT "         have been refit by Dickson and Millero (1987) on the 
seawater scale. "
        PRINT "    GEOSECS and Peng et al used the fit given in Mehrbach et al. 
"
        PRINT "    For freshwater, Millero (1979) refit data from Harned and 
Davis "
        PRINT "         for K1 and and Harned and Scholes for K2. "
        PRINT
        PRINT "    The following are approximate 2s PRECISIONs of the fits of 
the data: "
        PRINT "         (REMEMBER THAT PRECISION AND ACCURACY ARE NOT THE 
SAME!): "
        PRINT "                                       K1       K2 "
        PRINT "                                       ----    ---- "
        PRINT "         Roy                           2%      1.5% "
        PRINT "         Goyet and Poisson             2.5%    4.5% "
        PRINT "         Hansson, refit by DM          3%      4% "
        PRINT "         Mehrbach, refit by DM         2.5%    4.5% "
        PRINT "         DM combined fit               4%      6% "
        PRINT "         Mehrbach's fit                1.2%    2% "
        PRINT "         freshwater choice              .5%     .7% "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <P> to go to the program, <enter> to 
continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutConstantsp2:
        IF Q$ = "P" OR Q$ = "p" THEN EXIT SUB
'
'
```

```
'***************************************
AboutConstantsp4:
        CLS
        PRINT "    References are: "
        PRINT
        PRINT "         Roy, et al, Marine Chemistry 44:249-267,1993 "
        PRINT "            see also: Erratum, Marine Chemistry 45:337, 1994 "
        PRINT "            and Erratum, Marine Chemistry 52:183, 1996 "
        PRINT "         Goyet and Poisson, Deep-Sea Research 36:1635-1654, 1989
"
        PRINT "         Hansson, Deep-Sea Research 20:461-478, 1973 "
        PRINT "         Hansson, Acta Chemica Scandanavia, 27:931-944, 1973, "
        PRINT "         Mehrbach et al, Limnology and Oceaneanography, 18:897-
907, 1973 "
        PRINT "         Dickson and Millero, Deep-Sea Research, 34:1733-
1743,1987 "
        PRINT "            see also Corrigenda, Deep-Sea Research, 36:983, 1989
"
        PRINT "         Millero, F. J., Geochemica et Cosmochemica Acta 43:1651-
1661, 1979 "
        PRINT "         Harned and Davis, J American Chemical Society, 65:2030-
2037, 1943 "
        PRINT "         Harned and Scholes, J American Chemical Society,
43:1706-1709, 1941 "
        PRINT
        PRINT "    A very useful reference for all aspects of the CO2 system in
seawater is "
        PRINT "         DOE (1994), Handbook of methods for the analysis of the
various "
        PRINT "         parameters of the carbon dioxide system in sea water;
version 2, "
        PRINT "         A. G. Dickson and C. Goyet, eds. ORNL/CDIAC-74. "
        LOCATE 24, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Enter <B> to go back, <enter> to continue. ", Q$
        IF Q$ = "B" OR Q$ = "b" THEN GOTO AboutConstantsp3:
        Info$ = "Menu"
RETURN
'*****************************************************************************
END SUB
SUB CalculateAlkParts (pH, TC, K(), T(), HCO3, CO3, BAlk, OH, PAlk, SiAlk,
Hfree, HSO4, HF)
' SUB CalculateAlkParts, version 01.03, 10-10-97, written by Ernie Lewis.
' Inputs: pH, TC, K(), T()
' Outputs: HCO3, CO3, BAlk, OH, PAlk, SiAlk, Hfree, HSO4, HF
' This calculates the various contributions to the alkalinity.
' Though it is coded for H on the total pH scale, for the pH values occuring
' in seawater (pH > 6) it will be equally valid on any pH scale (H terms
' negligible) as long as the K constants are on that scale.
'
'
        K1 = K(1): K2 = K(2): KW = K(3): KB = K(4): KF = K(5)
        KS = K(6): KP1 = K(7): KP2 = K(8): KP3 = K(9): KSi = K(10)
        TB = T(1): TF = T(2): TS = T(3): TP = T(4): TSi = T(5)
'
'
        H = 10! ^ (-pH)
        HCO3 = TC * K1 * H / (K1 * H + H * H + K1 * K2)
        CO3 = TC * K1 * K2 / (K1 * H + H * H + K1 * K2)
        BAlk = TB * KB / (KB + H)
        OH = KW / H
                PhosTop = KP1 * KP2 * H + 2! * KP1 * KP2 * KP3 - H * H * H
                PhosBot = H * H * H + KP1 * H * H + KP1 * KP2 * H + KP1 * KP2 *
KP3
        PAlk = TP * PhosTop / PhosBot
        ' this is good to better than .0006*TP:
                'PAlk = TP*(-H/(KP1+H) + KP2/(KP2+H) + KP3/(KP3+H))
        SiAlk = TSi * KSi / (KSi + H)
        FREEtoTOT = (1! + TS / KS): ' pH scale conversion factor
        Hfree = H / FREEtoTOT: ' for H on the total scale
        HSO4 = TS / (1! + KS / Hfree): ' since KS is on the free scale
        HF = TF / (1! + KF / Hfree): ' since KF is on the free scale
END SUB
SUB CalculatefCO2fromTCpH (TC, pH, K0, K1, K2, fCO2)
' SUB CalculatefCO2fromTCpH, version 02.02, 12-13-96, written by Ernie Lewis.
' Inputs: TC, pH, K0, K1, K2
' Output: fCO2
' This calculates fCO2 from TC and pH, using K0, K1, and K2.
'
'
        H = 10! ^ (-pH)
        fCO2 = TC * H * H / (H * H + K1 * H + K1 * K2) / K0
END SUB
SUB CalculatepHfromTAfCO2 (TA, fCO2, K0, K(), T(), pH)
' SUB CalculatepHfromTAfCO2, version 04.01, 10-13-97, written by Ernie Lewis.
' Inputs: TA, fCO2, K0, K(), T()
' Output: pH
' This calculates pH from TA and fCO2 using K1 and K2 by Newton's method.
' It tries to solve for the pH at which Residual = 0.
' The starting guess is pH = 8.
' Though it is coded for H on the total pH scale, for the pH values occuring
' in seawater (pH > 6) it will be equally valid on any pH scale (H terms
' negligible) as long as the K constants are on that scale.
'
'
        K1 = K(1): K2 = K(2): KW = K(3): KB = K(4): KF = K(5)
        KS = K(6): KP1 = K(7): KP2 = K(8): KP3 = K(9): KSi = K(10)
        TB = T(1): TF = T(2): TS = T(3): TP = T(4): TSi = T(5)
'
'
        pHGuess = 8!: ' this is the first guess
        pHTol = .0001: ' this is .0001 pH units
        ln10 = LOG(10!)
        pH = pHGuess
        DO
                H = 10! ^ (-pH)
                HCO3 = K0 * K1 * fCO2 / H
                CO3 = K0 * K1 * K2 * fCO2 / (H * H)
                CAlk = HCO3 + 2! * CO3
                BAlk = TB * KB / (KB + H)
                OH = KW / H
                        PhosTop = KP1 * KP2 * H + 2! * KP1 * KP2 * KP3 - H * H *
H
```

```
                        PhosBot = H * H * H + KP1 * H * H + KP1 * KP2 * H + KP1
* KP2 * KP3
                PAlk = TP * PhosTop / PhosBot
                SiAlk = TSi * KSi / (KSi + H)
                FREEtoTOT = (1! + TS / KS): ' pH scale conversion factor
                Hfree = H / FREEtoTOT: ' for H on the total scale
                HSO4 = TS / (1! + KS / Hfree): ' since KS is on the free scale
                HF = TF / (1! + KF / Hfree): ' since KF is on the free scale
                Residual = TA - CAlk - BAlk - OH - PAlk - SiAlk + Hfree + HSO4 +
HF
'
'               find Slope dTA/dpH
'               (this is not exact, but keeps all important terms):
                Slope = ln10 * (HCO3 + 4! * CO3 + BAlk * H / (KB + H) + OH + H)
                deltapH = Residual / Slope: ' this is Newton's method
                ' to keep the jump from being too big:
                DO WHILE ABS(deltapH) > 1!: deltapH = deltapH / 2!: LOOP
                pH = pH + deltapH
        LOOP WHILE ABS(deltapH) > pHTol
END SUB
SUB CalculatepHfromTATC (TA, TC, K(), T(), pH)
' SUB CalculatepHfromTATC, version 04.01, 10-13-96, written by Ernie Lewis.
' Inputs: TA, TC, K(), T()
' Output: pH
' This calculates pH from TA and TC using K1 and K2 by Newton's method.
' It tries to solve for the pH at which Residual = 0.
' The starting guess is pH = 8.
' Though it is coded for H on the total pH scale, for the pH values occuring
' in seawater (pH > 6) it will be equally valid on any pH scale (H terms
' negligible) as long as the K constants are on that scale.
'
'
        K1 = K(1): K2 = K(2): KW = K(3): KB = K(4): KF = K(5)
        KS = K(6): KP1 = K(7): KP2 = K(8): KP3 = K(9): KSi = K(10)
        TB = T(1): TF = T(2): TS = T(3): TP = T(4): TSi = T(5)
'
'
        pHGuess = 8!: ' this is the first guess
        pHTol = .0001: ' this is .0001 pH units
        ln10 = LOG(10!)
        pH = pHGuess
        DO
                H = 10! ^ (-pH)
                Denom = (H * H + K1 * H + K1 * K2)
                CAlk = TC * K1 * (H + 2! * K2) / Denom
                BAlk = TB * KB / (KB + H)
                OH = KW / H
                        PhosTop = KP1 * KP2 * H + 2! * KP1 * KP2 * KP3 - H * H *
H
                        PhosBot = H * H * H + KP1 * H * H + KP1 * KP2 * H + KP1
* KP2 * KP3
                PAlk = TP * PhosTop / PhosBot
                SiAlk = TSi * KSi / (KSi + H)
                FREEtoTOT = (1! + TS / KS): ' pH scale conversion factor
                Hfree = H / FREEtoTOT: ' for H on the total scale
                HSO4 = TS / (1! + KS / Hfree): ' since KS is on the free scale
                HF = TF / (1! + KF / Hfree): ' since KF is on the free scale
                Residual = TA - CAlk - BAlk - OH - PAlk - SiAlk + Hfree + HSO4 +
HF
'
'               find Slope dTA/dpH:
'               (this is not exact, but keeps all important terms):
                Slope = ln10 * (TC * K1 * H * (H * H + K1 * K2 + 4! * H * K2) /
Denom / Denom + BAlk * H / (KB + H) + OH + H)
                deltapH = Residual / Slope: ' this is Newton's method
                ' to keep the jump from being too big:
                DO WHILE ABS(deltapH) > 1!: deltapH = deltapH / 2!: LOOP
                pH = pH + deltapH
        LOOP WHILE ABS(deltapH) > pHTol
END SUB
SUB CalculatepHfromTCfCO2 (TC, fCO2, K0, K1, K2, pH)
' SUB CalculatepHfromTCfCO2, version 02.02, 11-12-96, written by Ernie Lewis.
' Inputs: TC, fCO2, K0, K1, K2
' Output: pH
' This calculates pH from TC and fCO2 using K0, K1, and K2 by solving the
'       quadratic in H: fCO2 * K0 = TC * H * H / (K1 * H + H * H + K1 * K2).
' If there is not a real root, then pH is returned as -999.
'
'
        RR = K0 * fCO2 / TC
        IF RR >= 1 THEN pH = -999!: EXIT SUB
        ' check after sub to see if pH = -999.
        Discr = (K1 * RR) * (K1 * RR) + 4! * (1! - RR) * (K1 * K2 * RR)
        H = .5 * (K1 * RR + SQR(Discr)) / (1! - RR)
        pH = LOG(H) / LOG(.1)
END SUB
SUB CalculateTAfromTCpH (TC, pH, K(), T(), TA)
' SUB CalculateTAfromTCpH, version 02.02, 10-10-97, written by Ernie Lewis.
' Inputs: TC, pH, K(), T()
' Output: TA
' This calculates TA from TC and pH.
' Though it is coded for H on the total pH scale, for the pH values occuring
' in seawater (pH > 6) it will be equally valid on any pH scale (H terms
' negligible) as long as the K constants are on that scale.
'
'
        K1 = K(1): K2 = K(2): KW = K(3): KB = K(4): KF = K(5)
        KS = K(6): KP1 = K(7): KP2 = K(8): KP3 = K(9): KSi = K(10)
        TB = T(1): TF = T(2): TS = T(3): TP = T(4): TSi = T(5)
'
'
        H = 10! ^ (-pH)
        CAlk = TC * K1 * (H + 2! * K2) / (H * H + K1 * H + K1 * K2)
        BAlk = TB * KB / (KB + H)
        OH = KW / H
                PhosTop = KP1 * KP2 * H + 2! * KP1 * KP2 * KP3 - H * H * H
                PhosBot = H * H * H + KP1 * H * H + KP1 * KP2 * H + KP1 * KP2 *
KP3
        PAlk = TP * PhosTop / PhosBot
        SiAlk = TSi * KSi / (KSi + H)
        FREEtoTOT = (1! + TS / KS): ' pH scale conversion factor
        Hfree = H / FREEtoTOT: ' for H on the total scale
        HSO4 = TS / (1! + KS / Hfree): ' since KS is on the free scale
        HF = TF / (1! + KF / Hfree): ' since KF is on the free scale
        TA = CAlk + BAlk + OH + PAlk + SiAlk - Hfree - HSO4 - HF
END SUB
SUB CalculateTCfrompHfCO2 (pH, fCO2, K0, K1, K2, TC)
' SUB CalculateTCfrompHfCO2, version 01.02, 12-13-96, written by Ernie Lewis.
' Inputs: pH, fCO2, K0, K1, K2
' Output: TC
' This calculates TC from pH and fCO2, using K0, K1, and K2.
'
'
        H = 10! ^ (-pH)
        TC = K0 * fCO2 * (H * H + K1 * H + K1 * K2) / (H * H)
END SUB
SUB CalculateTCfromTApH (TA, pH, K(), T(), TC)
' SUB CalculateTCfromTApH, version 02.03, 10-10-97, written by Ernie Lewis.
' Inputs: TA, pH, K(), T()
' Output: TC
' This calculates TC from TA and pH.
' Though it is coded for H on the total pH scale, for the pH values occuring
' in seawater (pH > 6) it will be equally valid on any pH scale (H terms
' negligible) as long as the K constants are on that scale.
'
'
        K1 = K(1): K2 = K(2): KW = K(3): KB = K(4): KF = K(5)
        KS = K(6): KP1 = K(7): KP2 = K(8): KP3 = K(9): KSi = K(10)
        TB = T(1): TF = T(2): TS = T(3): TP = T(4): TSi = T(5)
'
'
        H = 10! ^ (-pH)
        BAlk = TB * KB / (KB + H)
        OH = KW / H
                PhosTop = KP1 * KP2 * H + 2! * KP1 * KP2 * KP3 - H * H * H
                PhosBot = H * H * H + KP1 * H * H + KP1 * KP2 * H + KP1 * KP2 *
KP3
        PAlk = TP * PhosTop / PhosBot
        SiAlk = TSi * KSi / (KSi + H)
        FREEtoTOT = (1! + TS / KS): ' pH scale conversion factor
        Hfree = H / FREEtoTOT: ' for H on the total scale
        HSO4 = TS / (1! + KS / Hfree): ' since KS is on the free scale
        HF = TF / (1! + KF / Hfree): ' since KF is on the free scale
        CAlk = TA - BAlk - OH - PAlk - SiAlk + Hfree + HSO4 + HF
        TC = CAlk * (H * H + K1 * H + K1 * K2) / (K1 * (H + 2! * K2))
END SUB
SUB Case1Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal, K(), T(),
TempCinp, TempCout, Pdbarinp, Pdbarout, TA, TC, pHinp, fCO2inp, pCO2inp, pHout,
fCO2out, pCO2out)
' SUB Case1Partials, version 01.04, 03-12-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, fORp$
' Inputs: Sal, K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout
' Inputs: TA, TC, pHinp, fCO2inp, pCO2inp, pHout, fCO2out, pCO2out
' Outputs: none
' This calculates and prints the partials for case 1: input TA, TC.
'
'
        TA0 = TA: TC0 = TC: Sal0 = Sal
        pHinp0 = pHinp: pHout0 = pHout
        fCO2inp0 = fCO2inp: pCO2inp0 = pCO2inp
        fCO2out0 = fCO2out: pCO2out0 = pCO2out
        CALL SetParametersForPartials(dTA, dTC, dpH, dfCO2, dSal, dTempC,
dPdbar, pcdK0, pcdK1, pcdK2)
' **************
'Increase TA by dTA
        TA = TA0 + dTA
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                GOSUB CalculateStuffForCase1Partials:
                dpHinpdTA = (pH - pHinp0) / dTA
                dfCO2inpdTA = (fCO2 - fCO2inp0) / dTA
                dpCO2inpdTA = (pCO2 - pCO2inp0) / dTA
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTA = (pH - pHout0) / dTA
                dfCO2outdTA = (fCO2 - fCO2out0) / dTA
                dpCO2outdTA = (pCO2 - pCO2out0) / dTA
        TA = TA0
' **************
'Increase TC by dTC
        TC = TC0 + dTC
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                GOSUB CalculateStuffForCase1Partials:
                dpHinpdTC = (pH - pHinp0) / dTC
                dfCO2inpdTC = (fCO2 - fCO2inp0) / dTC
                dpCO2inpdTC = (pCO2 - pCO2inp0) / dTC
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTC = (pH - pHout0) / dTC
                dfCO2outdTC = (fCO2 - fCO2out0) / dTC
                dpCO2outdTC = (pCO2 - pCO2out0) / dTC
        TC = TC0
' **************
'Increase Sal by dSal
        Sal = Sal0 + dSal
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                GOSUB CalculateStuffForCase1Partials:
                dpHinpdSal = (pH - pHinp0) / dSal
                dfCO2inpdSal = (fCO2 - fCO2inp0) / dSal
                dpCO2inpdSal = (pCO2 - pCO2inp0) / dSal
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdSal = (pH - pHout0) / dSal
                dfCO2outdSal = (fCO2 - fCO2out0) / dSal
                dpCO2outdSal = (pCO2 - pCO2out0) / dSal
        Sal = Sal0
' **************
```

```
'Increase TempCinp by dTempC
'       Do at Tinp, Pinp
                TempC = TempCinp + dTempC: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                GOSUB CalculateStuffForCase1Partials:
                dpHinpdTempCinp = (pH - pHinp0) / dTempC
                dfCO2inpdTempCinp = (fCO2 - fCO2inp0) / dTempC
                dpCO2inpdTempCinp = (pCO2 - pCO2inp0) / dTempC
'       Output results not affected by changes in Tinp in this case
' **************
'Increase Pdbar by dPdbar
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp + dPdbar
                GOSUB GetConstantsforCase1Partials:
                GOSUB CalculateStuffForCase1Partials:
                dpHinpdPdbarinp = (pH - pHinp0) / dPdbar
                dfCO2inpdPdbarinp = (fCO2 - fCO2inp0) / dPdbar
                dpCO2inpdPdbarinp = (pCO2 - pCO2inp0) / dPdbar
'       Output results not affected by changes in Pinp in this case
' **************
'Increase K0 by pcdK0 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                K0 = K0 * (1! + pcdK0 / 100!)
                GOSUB CalculateStuffForCase1Partials:
                ' pH doesn't depend on K0
                dfCO2inppcdK0 = (fCO2 - fCO2inp0) / pcdK0
                dpCO2inppcdK0 = (pCO2 - pCO2inp0) / pcdK0
'       Output results not affected by changes in K0 at input conditions in this
case
'***************
'Increase K1 by pcdK1 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                K(1) = K(1) * (1! + pcdK1 / 100!): K1 = K(1)
                GOSUB CalculateStuffForCase1Partials:
                dpHinppcdK1 = (pH - pHinp0) / pcdK1
                dfCO2inppcdK1 = (fCO2 - fCO2inp0) / pcdK1
                dpCO2inppcdK1 = (pCO2 - pCO2inp0) / pcdK1
'       Output results not affected by changes in K1 at input conditions in this
case
' **************
'Increase K2 by pcdK2 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase1Partials:
                K(2) = K(2) * (1! + pcdK2 / 100!): K2 = K(2)
                GOSUB CalculateStuffForCase1Partials:
                dpHinppcdK2 = (pH - pHinp0) / pcdK2
                dfCO2inppcdK2 = (fCO2 - fCO2inp0) / pcdK2
                dpCO2inppcdK2 = (pCO2 - pCO2inp0) / pcdK2
'       Output results not affected by changes in K2 at input conditions in this
case
'
'
```

```
' ******************************************
PrintPartialsForCase1:
        AAA$ = "#.####  ####.#           #.####  ####.# "
        IF fORp$ = "f" THEN
                PRINT USING "                           \    \   fCO2
\    \   fCO2 "; pHScale$; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; pHinp0;
fCO2inp0 * 1000000!; pHout0; fCO2out0 * 1000000!
        ELSEIF fORp$ = "p" THEN
                PRINT USING "                           \    \   pCO2
\    \   pCO2 "; pHScale$; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; pHinp0;
pCO2inp0 * 1000000!; pHout0; pCO2out0 * 1000000!
        END IF
        PRINT USING "    1 umol/kg in TA        " + AAA$; dpHinpdTA / 1000000!;
dfCO2inpdTA; dpHoutdTA / 1000000!; dfCO2outdTA
        PRINT USING "    1 umol/kg in TC        " + AAA$; dpHinpdTC / 1000000!;
dfCO2inpdTC; dpHoutdTC / 1000000!; dfCO2outdTC
        IF WhichKs% <> 8 THEN
                PRINT USING "    1 in salinity          " + AAA$; dpHinpdSal;
dfCO2inpdSal * 1000000!; dpHoutdSal; dfCO2outdSal * 1000000!
        END IF
        PRINT USING "    1 deg C in input T     #.####  ####.# ";
dpHinpdTempCinp; dfCO2inpdTempCinp * 1000000!
        PRINT USING "    100 dbar in input P    #.####  ####.# ";
dpHinpdPdbarinp * 100!; dfCO2inpdPdbarinp * 1000000! * 100!
        PRINT USING "    1% K0 at input T               ####.# "; dfCO2inppcdK0
* 1000000!
        PRINT USING "    1% K1 at input T, P    #.####  ####.# "; dpHinppcdK1;
dfCO2inppcdK1 * 1000000!
        PRINT USING "    1% K2 at input T, P    #.####  ####.# "; dpHinppcdK2;
dfCO2inppcdK2 * 1000000!
EXIT SUB
'****************************************************************************
GetConstantsforCase1Partials:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
RETURN
CalculateStuffForCase1Partials:
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatepHfromTATC(TA, TC, K(), T(), pH)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 = fCO2 /
FugFac
RETURN
END SUB
SUB Case2Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal, K(), T(),
TempCinp, TempCout, Pdbarinp, Pdbarout, TA, pHinp, TC, fCO2inp, pCO2inp, pHout,
fCO2out, pCO2out)
' SUB Case2Partials, version 01.04, 03-12-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, fORp$
' Inputs: Sal, K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout
' Inputs: TA, pHinp, TC, fCO2inp, pCO2inp, pHout, fCO2out, pCO2out
' Outputs: none
' This calculates and prints the partials for case 2: input TA, pH.
'
'
        TA0 = TA: TC0 = TC: Sal0 = Sal
        pHinp0 = pHinp: pHout0 = pHout
        fCO2inp0 = fCO2inp: pCO2inp0 = pCO2inp
        fCO2out0 = fCO2out: pCO2out0 = pCO2out
        CALL SetParametersForPartials(dTA, dTC, dpH, dfCO2, dSal, dTempC,
dPdbar, pcdK0, pcdK1, pcdK2)
' **************
'Increase TA by dTA
        TA = TA0 + dTA
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                dTCdTA = (TC - TC0) / dTA
                dfCO2inpdTA = (fCO2 - fCO2inp0) / dTA
                dpCO2inpdTA = (pCO2 - pCO2inp0) / dTA
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTA = (pH - pHout0) / dTA
                dfCO2outdTA = (fCO2 - fCO2out0) / dTA
                dpCO2outdTA = (pCO2 - pCO2out0) / dTA
        TA = TA0
' **************
'Increase pH by dpH (this is pH at input conditions)
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                pH = pHinp0 + dpH
                GOSUB CalculateStuffForCase2Partials:
                dTCdpH = (TC - TC0) / dpH
                dfCO2inpdpH = (fCO2 - fCO2inp0) / dpH
                dpCO2inpdpH = (pCO2 - pCO2inp0) / dpH
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdpH = (pH - pHout0) / dpH
                dfCO2outdpH = (fCO2 - fCO2out0) / dpH
                dpCO2outdpH = (pCO2 - pCO2out0) / dpH
' **************
'Increase Sal by dSal
        Sal = Sal0 + dSal
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                dTCdSal = (TC - TC0) / dSal
                dfCO2inpdSal = (fCO2 - fCO2inp0) / dSal
                dpCO2inpdSal = (pCO2 - pCO2inp0) / dSal
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdSal = (pH - pHout0) / dSal
                dfCO2outdSal = (fCO2 - fCO2out0) / dSal
                dpCO2outdSal = (pCO2 - pCO2out0) / dSal
        Sal = Sal0
' **************
'Increase TempCinp by dTempC
'       Do at Tinp, Pinp
                TempC = TempCinp + dTempC: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                dTCdTempCinp = (TC - TC0) / dTempC
                dfCO2inpdTempCinp = (fCO2 - fCO2inp0) / dTempC
                dpCO2inpdTempCinp = (pCO2 - pCO2inp0) / dTempC
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTempCinp = (pH - pHout0) / dTempC
                dfCO2outdTempCinp = (fCO2 - fCO2out0) / dTempC
                dpCO2outdTempCinp = (pCO2 - pCO2out0) / dTempC
' **************
'Increase Pdbarinp by dPdbar
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp + dPdbar
                GOSUB GetConstantsforCase2Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                dTCdPdbarinp = (TC - TC0) / dPdbar
                dfCO2inpdPdbarinp = (fCO2 - fCO2inp0) / dPdbar
                dpCO2inpdPdbarinp = (pCO2 - pCO2inp0) / dPdbar
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdPdbarinp = (pH - pHout0) / dPdbar
                dfCO2outdPdbarinp = (fCO2 - fCO2out0) / dPdbar
                dpCO2outdPdbarinp = (pCO2 - pCO2out0) / dPdbar
' **************
'Increase K0 by pcdK0 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                K0 = K0 * (1! + pcdK0 / 100!)
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                ' TC does not depend on K0
                dfCO2inppcdK0 = (fCO2 - fCO2inp0) / pcdK0
                dpCO2inppcdK0 = (pCO2 - pCO2inp0) / pcdK0
'       Output results not affected by changes in K0 at input conditions in this
case
' **************
```

```
'Increase K1 by pcdK1 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                K(1) = K(1) * (1! + pcdK1 / 100!): K1 = K(1)
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                dTCpcdK1 = (TC - TC0) / pcdK1
                dfCO2inppcdK1 = (fCO2 - fCO2inp0) / pcdK1
                dpCO2inppcdK1 = (pCO2 - pCO2inp0) / pcdK1
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK1 = (pH - pHout0) / pcdK1
                dfCO2outpcdK1 = (fCO2 - fCO2out0) / pcdK1
                dpCO2outpcdK1 = (pCO2 - pCO2out0) / pcdK1
' **************
'Increase K2 by pcdK2 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase2Partials:
                K(2) = K(2) * (1! + pcdK2 / 100!): K2 = K(2)
                pH = pHinp0
                GOSUB CalculateStuffForCase2Partials:
                dTCpcdK2 = (TC - TC0) / pcdK2
                dfCO2inppcdK2 = (fCO2 - fCO2inp0) / pcdK2
                dpCO2inppcdK2 = (pCO2 - pCO2inp0) / pcdK2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK2 = (pH - pHout0) / pcdK2
                dfCO2outpcdK2 = (fCO2 - fCO2out0) / pcdK2
                dpCO2outpcdK2 = (pCO2 - pCO2out0) / pcdK2
'
'
' ******************************************
        AAA$ = "####.#  ####.#           #.####  ####.# "
PrintPartialsForCase2:
        IF fORp$ = "f" THEN
                PRINT USING "                             TC     fCO2
\    \   fCO2 "; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TC0 *
1000000!; fCO2inp0 * 1000000!; pHout0; fCO2out0 * 1000000!
        ELSEIF fORp$ = "p" THEN
                PRINT USING "                             TC     pCO2
\    \   pCO2 "; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TC0 *
1000000!; pCO2inp0 * 1000000!; pHout0; pCO2out0 * 1000000!
        END IF
        PRINT USING "    1 umol/kg in TA        " + AAA$; dTCdTA; dfCO2inpdTA;
dpHoutdTA / 1000000!; dfCO2outdTA
```

```
        PRINT USING "    .001 in input pH       " + AAA$; dTCdpH * 1000000! /
1000!; dfCO2inpdpH * 1000000! / 1000!; dpHoutdpH / 1000!; dfCO2outdpH * 1000000!
/ 1000!
        IF WhichKs% <> 8 THEN
                PRINT USING "    1 in salinity          " + AAA$; dTCdSal *
1000000!; dfCO2inpdSal * 1000000!; dpHoutdSal; dfCO2outdSal * 1000000!
        END IF
        PRINT USING "    1 deg C in input T     " + AAA$; dTCdTempCinp *
1000000!; dfCO2inpdTempCinp * 1000000!; dpHoutdTempCinp; dfCO2outdTempCinp *
1000000!
        PRINT USING "    100 dbar in input P    " + AAA$; dTCdPdbarinp *
1000000! * 100!; dfCO2inpdPdbarinp * 1000000! * 100!; dpHoutdPdbarinp * 100!;
dfCO2outdPdbarinp * 1000000! * 100!
        PRINT USING "    1% K0 at input T               ####.# "; dfCO2inppcdK0
* 1000000!
        PRINT USING "    1% K1 at input T, P    " + AAA$; dTCpcdK1 * 1000000!;
dfCO2inppcdK1 * 1000000!; dpHoutpcdK1; dfCO2outpcdK1 * 1000000!
        PRINT USING "    1% K2 at input T, P    " + AAA$; dTCpcdK2 * 1000000!;
dfCO2inppcdK2 * 1000000!; dpHoutpcdK2; dfCO2outpcdK2 * 1000000!
'
'
        TC = TC0: ' to pass back the value that came in
EXIT SUB
'****************************************************************************
GetConstantsforCase2Partials:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
RETURN
CalculateStuffForCase2Partials:
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculateTCfromTApH(TA, pH, K(), T(), TC)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 = fCO2 /
FugFac
RETURN
END SUB
SUB Case3Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal, K(), T(),
TempCinp, TempCout, Pdbarinp, Pdbarout, TA, fCO2inp, pCO2inp, TC, pHinp, pHout,
fCO2out, pCO2out)
' SUB Case3Partials, version 01.04, 03-12-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, fORp$
' Inputs: Sal, K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout
' Inputs: TA, fCO2inp, pCO2inp, TC, pHinp, pHout, fCO2out, pCO2out
' Outputs: none
' This calculates and prints the partials for case 3: input TA, fCO2 or pCO2.
'
'
        TA0 = TA: TC0 = TC: Sal0 = Sal
        pHinp0 = pHinp: pHout0 = pHout
        fCO2inp0 = fCO2inp
        fCO2out0 = fCO2out: pCO2out0 = pCO2out
        CALL SetParametersForPartials(dTA, dTC, dpH, dfCO2, dSal, dTempC,
dPdbar, pcdK0, pcdK1, pcdK2)
' *************
'Increase TA by dTA
        TA0 = TA: TA = TA0 + dTA
```

```
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase3Partials:
                dTCdTA = (TC - TC0) / dTA
                dpHinpdTA = (pH - pHinp0) / dTA
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTA = (pH - pHout0) / dTA
                dfCO2outdTA = (fCO2 - fCO2out0) / dTA
                dpCO2outdTA = (pCO2 - pCO2out0) / dTA
        TA = TA0
' **************
'Increase fCO2 by dfCO2 (this is fCO2 at input conditions)
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
'               this must be after call to constants for correct FugFac
                dpCO2 = dfCO2 / FugFac
                fCO2 = fCO2inp0 + dfCO2
                GOSUB CalculateStuffForCase3Partials:
                dTCdfCO2 = (TC - TC0) / dfCO2
                dTCdpCO2 = (TC - TC0) / dpCO2
                dpHinpdfCO2 = (pH - pHinp0) / dfCO2
                dpHinpdpCO2 = (pH - pHinp0) / dpCO2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdfCO2 = (pH - pHout0) / dfCO2
                dpHoutdpCO2 = (pH - pHout0) / dpCO2
                dfCO2outdfCO2 = (fCO2 - fCO2out0) / dfCO2
                dpCO2outdpCO2 = (pCO2 - pCO2out0) / dpCO2
' **************
'Increase Sal by dSal
        Sal = Sal0 + dSal
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
                fCO2 = fCO2inp0
'               since conversion of pCO2 to fCO2 depends on Sal, Temp:
                IF fORp$ = "p" THEN fCO2 = pCO2inp * FugFac
                GOSUB CalculateStuffForCase3Partials:
                dTCdSal = (TC - TC0) / dSal
                dpHinpdSal = (pH - pHinp0) / dSal
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdSal = (pH - pHout0) / dSal
                dfCO2outdSal = (fCO2 - fCO2out0) / dSal
                dpCO2outdSal = (pCO2 - pCO2out0) / dSal
        Sal = Sal0
' **************
```

```
'Increase TempCinp by dTempC
'       Do at Tinp, Pinp
                TempC = TempCinp + dTempC: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
                fCO2 = fCO2inp0
'               since conversion of pCO2 to fCO2 depends on Sal, Temp:
                IF fORp$ = "p" THEN fCO2 = pCO2inp * FugFac
                GOSUB CalculateStuffForCase3Partials:
                dTCdTempCinp = (TC - TC0) / dTempC
                dpHinpdTempCinp = (pH - pHinp0) / dTempC
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTempCinp = (pH - pHout0) / dTempC
                dfCO2outdTempCinp = (fCO2 - fCO2out0) / dTempC
                dpCO2outdTempCinp = (pCO2 - pCO2out0) / dTempC
' **************
'Increase Pdbarinp by dPdbar
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp + dPdbar
                GOSUB GetConstantsforCase3Partials:
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase3Partials:
                dTCdPdbarinp = (TC - TC0) / dPdbar
                dpHinpdPdbarinp = (pH - pHinp0) / dPdbar
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdPdbarinp = (pH - pHout0) / dPdbar
                dfCO2outdPdbarinp = (fCO2 - fCO2out0) / dPdbar
                dpCO2outdPdbarinp = (pCO2 - pCO2out0) / dPdbar
' **************
'Increase K0 by pcdK0 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
                K0 = K0 * (1! + pcdK0 / 100!)
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase3Partials:
                dTCpcdK0 = (TC - TC0) / pcdK0
                dpHinppcdK0 = (pH - pHinp0) / pcdK0
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK0 = (pH - pHout0) / pcdK0
                dfCO2outpcdK0 = (fCO2 - fCO2out0) / pcdK0
                dpCO2outpcdK0 = (pCO2 - pCO2out0) / pcdK0
' **************
'Increase K1 by pcdK1 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
                K(1) = K(1) * (1! + pcdK1 / 100!): K1 = K(1)
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase3Partials:
                dpHinppcdK1 = (pH - pHinp0) / pcdK1
                dTCpcdK1 = (TC - TC0) / pcdK1
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK1 = (pH - pHout0) / pcdK1
                dfCO2outpcdK1 = (fCO2 - fCO2out0) / pcdK1
                dpCO2outpcdK1 = (pCO2 - pCO2out0) / pcdK1
' **************
'Increase K2 by pcdK2 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase3Partials:
                K(2) = K(2) * (1! + pcdK2 / 100!): K2 = K(2)
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase3Partials:
                dpHinppcdK2 = (pH - pHinp0) / pcdK2
                dTCpcdK2 = (TC - TC0) / pcdK2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK2 = (pH - pHout0) / pcdK2
                dfCO2outpcdK2 = (fCO2 - fCO2out0) / pcdK2
                dpCO2outpcdK2 = (pCO2 - pCO2out0) / pcdK2
'
'
' ******************************************
        AAA$ = "####.#  #.####           #.####  ####.# "
PrintPartialsForCase3:
        IF fORp$ = "f" THEN
                PRINT USING "                             TC    \    \
\    \   fCO2 "; pHScale$; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TC0 *
1000000!; pHinp0; pHout0; fCO2out0 * 1000000!
                PRINT USING "    1 umol/kg in TA        " + AAA$; dTCdTA;
dpHinpdTA / 1000000!; dpHoutdTA / 1000000!; dfCO2outdTA
                PRINT USING "    1 uatm in input fCO2   " + AAA$; dTCdfCO2;
dpHinpdfCO2 / 1000000!; dpHoutdfCO2 / 1000000!; dfCO2outdfCO2
        ELSEIF fORp$ = "p" THEN
                PRINT USING "                             TC    \    \
\    \   pCO2 "; pHScale$; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TC0 *
1000000!; pHinp0; pHout0; pCO2out0 * 1000000!
                PRINT USING "    1 umol/kg in TA        " + AAA$; dTCdTA;
dpHinpdTA / 1000000!; dpHoutdTA / 1000000!; dpCO2outdTA
                PRINT USING "    1 uatm in input pCO2   " + AAA$; dTCdpCO2;
dpHinpdpCO2 / 1000000!; dpHoutdpCO2 / 1000000!; dpCO2outdpCO2
        END IF
        IF WhichKs% <> 8 THEN
```

```
                PRINT USING "    1 in salinity          " + AAA$; dTCdSal *
1000000!; dpHinpdSal; dpHoutdSal; dfCO2outdSal * 1000000!
        END IF
        PRINT USING "    1 deg C in input T     " + AAA$; dTCdTempCinp *
1000000!; dpHinpdTempCinp; dpHoutdTempCinp; dfCO2outdTempCinp * 1000000!
        PRINT USING "    100 dbar in input P    " + AAA$; dTCdPdbarinp *
1000000! * 100!; dpHinpdPdbarinp * 100!; dpHoutdPdbarinp * 100!;
dfCO2outdPdbarinp * 1000000! * 100!
        PRINT USING "    1% K0 at input T       " + AAA$; dTCpcdK0 * 1000000!;
dpHinppcdK0; dpHoutpcdK0; dfCO2outpcdK0 * 1000000!
        PRINT USING "    1% K1 at input T, P    " + AAA$; dTCpcdK1 * 1000000!;
dpHinppcdK1; dpHoutpcdK1; dfCO2outpcdK1 * 1000000!
        PRINT USING "    1% K2 at input T, P    " + AAA$; dTCpcdK2 * 1000000!;
dpHinppcdK2; dpHoutpcdK2; dfCO2outpcdK2 * 1000000!
'
'
        TC = TC0: ' to pass back the value that came in
EXIT SUB
'****************************************************************************
GetConstantsforCase3Partials:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
RETURN
CalculateStuffForCase3Partials:
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatepHfromTAfCO2(TA, fCO2, K0, K(), T(), pH)
        CALL CalculateTCfromTApH(TA, pH, K(), T(), TC)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
RETURN
END SUB
SUB Case4Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal, K(), T(),
TempCinp, TempCout, Pdbarinp, Pdbarout, TC, pHinp, TA, fCO2inp, pCO2inp, pHout,
fCO2out, pCO2out)
' SUB Case4Partials, version 01.04, 03-12-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, fORp$
' Inputs: Sal, K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout
' Inputs: TC, pHinp, TA, fCO2inp, pCO2inp, pHout, fCO2out, pCO2out
' Outputs: none
' This calculates and prints the partials for case 4: input TC, pH.
'
'
        TA0 = TA: TC0 = TC: Sal0 = Sal
        pHinp0 = pHinp: pHout0 = pHout
        fCO2inp0 = fCO2inp: pCO2inp0 = pCO2inp
        fCO2out0 = fCO2out: pCO2out0 = pCO2out
        CALL SetParametersForPartials(dTA, dTC, dpH, dfCO2, dSal, dTempC,
dPdbar, pcdK0, pcdK1, pcdK2)
' **************
'Increase TC by dTC
        TC = TC0 + dTC
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                dTAdTC = (TA - TA0) / dTC
                pCO2 = fCO2 / FugFac
                dfCO2inpdTC = (fCO2 - fCO2inp0) / dTC
                dpCO2inpdTC = (pCO2 - pCO2inp0) / dTC
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTC = (pH - pHout0) / dTC
                dfCO2outdTC = (fCO2 - fCO2out0) / dTC
                dpCO2outdTC = (pCO2 - pCO2out0) / dTC
        TC = TC0
' **************
'Increase pH by dpH (this is pH at input conditions)
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                pH = pHinp0 + dpH
                GOSUB CalculateStuffForCase4Partials:
                dTAdpH = (TA - TA0) / dpH
                dfCO2inpdpH = (fCO2 - fCO2inp0) / dpH
                dpCO2inpdpH = (pCO2 - pCO2inp0) / dpH
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdpH = (pH - pHout0) / dpH
                dfCO2outdpH = (fCO2 - fCO2out0) / dpH
                dpCO2outdpH = (pCO2 - pCO2out0) / dpH
' **************
'Increase Sal by dSal
        Sal = Sal0 + dSal
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                dTAdSal = (TA - TA0) / dSal
                dfCO2inpdSal = (fCO2 - fCO2inp0) / dSal
                dpCO2inpdSal = (pCO2 - pCO2inp0) / dSal
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdSal = (pH - pHout0) / dSal
                dfCO2outdSal = (fCO2 - fCO2out0) / dSal
                dpCO2outdSal = (pCO2 - pCO2out0) / dSal
        Sal = Sal0
' **************
'Increase TempCinp by dTempC
'       Do at Tinp, Pinp
                TempC = TempCinp + dTempC: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                dTAdTempCinp = (TA - TA0) / dTempC
                dfCO2inpdTempCinp = (fCO2 - fCO2inp0) / dTempC
                dpCO2inpdTempCinp = (pCO2 - pCO2inp0) / dTempC
```

```
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTempCinp = (pH - pHout0) / dTempC
                dfCO2outdTempCinp = (fCO2 - fCO2out0) / dTempC
                dpCO2outdTempCinp = (pCO2 - pCO2out0) / dTempC
' **************
'Increase Pdbarinp by dPdbar
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp + dPdbar
                GOSUB GetConstantsforCase4Partials:
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                dTAdPdbarinp = (TA - TA0) / dPdbar
                dfCO2inpdPdbarinp = (fCO2 - fCO2inp0) / dPdbar
                dpCO2inpdPdbarinp = (pCO2 - pCO2inp0) / dPdbar
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdPdbarinp = (pH - pHout0) / dPdbar
                dfCO2outdPdbarinp = (fCO2 - fCO2out0) / dPdbar
                dpCO2outdPdbarinp = (pCO2 - pCO2out0) / dPdbar
' **************
'Increase K0 by pcdK0 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                K0 = K0 * (1! + pcdK0 / 100!)
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                ' TA does not depend on K0 but I need TA ( = TA0) for output
                dfCO2inppcdK0 = (fCO2 - fCO2inp0) / pcdK0
                dpCO2inppcdK0 = (pCO2 - pCO2inp0) / pcdK0
'       Output results not affected by changes in K0 at input conditions in this
case
' **************
'Increase K1 by pcdK1 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                K(1) = K(1) * (1! + pcdK1 / 100!): K1 = K(1)
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                dTApcdK1 = (TA - TA0) / pcdK1
                dfCO2inppcdK1 = (fCO2 - fCO2inp0) / pcdK1
                dpCO2inppcdK1 = (pCO2 - pCO2inp0) / pcdK1
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK1 = (pH - pHout0) / pcdK1
                dfCO2outpcdK1 = (fCO2 - fCO2out0) / pcdK1
                dpCO2outpcdK1 = (pCO2 - pCO2out0) / pcdK1
' **************
'Increase K2 by pcdK2 % at input conditions only
```

```
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase4Partials:
                K(2) = K(2) * (1! + pcdK2 / 100!): K2 = K(2)
                pH = pHinp0
                GOSUB CalculateStuffForCase4Partials:
                dTApcdK2 = (TA - TA0) / pcdK2
                dfCO2inppcdK2 = (fCO2 - fCO2inp0) / pcdK2
                dpCO2inppcdK2 = (pCO2 - pCO2inp0) / pcdK2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK2 = (pH - pHout0) / pcdK2
                dfCO2outpcdK2 = (fCO2 - fCO2out0) / pcdK2
                dpCO2outpcdK2 = (pCO2 - pCO2out0) / pcdK2
'
'
' ********************************************
        AAA$ = "####.#  ####.#           #.####  ####.# "
PrintPartialsForCase4:
        IF fORp$ = "f" THEN
                PRINT USING "                             TA     fCO2
\    \   fCO2 "; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TA0 *
1000000!; fCO2inp0 * 1000000!; pHout0; fCO2out0 * 1000000!
        ELSEIF fORp$ = "p" THEN
                PRINT USING "                             TA     pCO2
\    \   pCO2 "; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TA0 *
1000000!; pCO2inp0 * 1000000!; pHout0; pCO2out0 * 1000000!
        END IF
        PRINT USING "    1 umol/kg in TC        " + AAA$; dTAdTC; dfCO2inpdTC;
dpHoutdTC / 1000000!; dfCO2outdTC
        PRINT USING "    .001 in input pH       " + AAA$; dTAdpH * 1000000! /
1000!; dfCO2inpdpH * 1000000! / 1000!; dpHoutdpH / 1000!; dfCO2outdpH * 1000000!
/ 1000!
        IF WhichKs% <> 8 THEN
                PRINT USING "    1 in salinity          " + AAA$; dTAdSal *
1000000!; dfCO2inpdSal * 1000000!; dpHoutdSal; dfCO2outdSal * 1000000!
        END IF
        PRINT USING "    1 deg C in input T     " + AAA$; dTAdTempCinp *
1000000!; dfCO2inpdTempCinp * 1000000!; dpHoutdTempCinp; dfCO2outdTempCinp *
1000000!
        PRINT USING "    100 dbar in input P    " + AAA$; dTAdPdbarinp *
1000000! * 100!; dfCO2inpdPdbarinp * 1000000! * 100!; dpHoutdPdbarinp * 100!;
dfCO2outdPdbarinp * 1000000! * 100!
        PRINT USING "    1% K0 at input T               ####.# "; dfCO2inppcdK0
* 1000000!
        PRINT USING "    1% K1 at input T, P    " + AAA$; dTApcdK1 * 1000000!;
dfCO2inppcdK1 * 1000000!; dpHoutpcdK1; dfCO2outpcdK1 * 1000000!
        PRINT USING "    1% K2 at input T, P    " + AAA$; dTApcdK2 * 1000000!;
dfCO2inppcdK2 * 1000000!; dpHoutpcdK2; dfCO2outpcdK2 * 1000000!
```

```
'
'
        TA = TA0: ' to pass back the value that came in
EXIT SUB
'****************************************************************************
GetConstantsforCase4Partials:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
RETURN
CalculateStuffForCase4Partials:
        CALL CalculateTAfromTCpH(TC, pH, K(), T(), TA)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 = fCO2 /
FugFac
RETURN
END SUB
SUB Case5Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal, K(), T(),
TempCinp, TempCout, Pdbarinp, Pdbarout, TC, fCO2inp, pCO2inp, TA, pHinp, pHout,
fCO2out, pCO2out, TCfCO2Flag%)
' SUB Case5Partials, version 01.04, 03-12-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, fORp$
' Inputs: Sal, K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout
' Inputs: TC, fCO2inp, pCO2inp, TA, pHinp, pHout, fCO2out, pCO2out
' Inputs: TCfCO2Flag%
' Outputs: TCfCO2Flag%
' This calculates and prints the partials for case 5: input TC, fCO2 or pCO2.
'
'
        TA0 = TA: TC0 = TC: Sal0 = Sal
        pHinp0 = pHinp: pHout0 = pHout
        fCO2inp0 = fCO2inp
        fCO2out0 = fCO2out: pCO2out0 = pCO2out
        CALL SetParametersForPartials(dTA, dTC, dpH, dfCO2, dSal, dTempC,
dPdbar, pcdK0, pcdK1, pcdK2)
' **************
'Increase TC by dTC
        TC = TC0 + dTC
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase5Partials:
                dpHinpdTC = (pH - pHinp0) / dTC
                dTAdTC = (TA - TA0) / dTC
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTC = (pH - pHout0) / dTC
                dfCO2outdTC = (fCO2 - fCO2out0) / dTC
                dpCO2outdTC = (pCO2 - pCO2out0) / dTC
        TC = TC0
' **************
'Increase fCO2 by dfCO2 (this is fCO2 at input conditions)
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
'               this must be after call to constants for correct FugFac
                dpCO2 = dfCO2 / FugFac
                fCO2 = fCO2inp0 + dfCO2
                GOSUB CalculateStuffForCase5Partials:
                dpHinpdfCO2 = (pH - pHinp0) / dfCO2
                dpHinpdpCO2 = (pH - pHinp0) / dpCO2
                dTAdfCO2 = (TA - TA0) / dfCO2
                dTAdpCO2 = (TA - TA0) / dpCO2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdfCO2 = (pH - pHout0) / dfCO2
                dpHoutdpCO2 = (pH - pHout0) / dpCO2
                dfCO2outdfCO2 = (fCO2 - fCO2out0) / dfCO2
                dpCO2outdpCO2 = (pCO2 - pCO2out0) / dpCO2
' **************
'Increase Sal by dSal
        Sal = Sal0 + dSal
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
                fCO2 = fCO2inp0
'               since conversion of pCO2 to fCO2 depends on Sal, Temp:
                IF fORp$ = "p" THEN fCO2 = pCO2inp * FugFac
                GOSUB CalculateStuffForCase5Partials:
                dTAdSal = (TA - TA0) / dSal
                dpHinpdSal = (pH - pHinp0) / dSal
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdSal = (pH - pHout0) / dSal
                dfCO2outdSal = (fCO2 - fCO2out0) / dSal
                dpCO2outdSal = (pCO2 - pCO2out0) / dSal
        Sal = Sal0
' **************
'Increase TempCinp by dTempC
'       Do at Tinp, Pinp
                TempC = TempCinp + dTempC: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
                fCO2 = fCO2inp0
'               since conversion of pCO2 to fCO2 depends on Sal, Temp:
                IF fORp$ = "p" THEN fCO2 = pCO2inp * FugFac
                GOSUB CalculateStuffForCase5Partials:
                dTAdTempCinp = (TA - TA0) / dTempC
                dpHinpdTempCinp = (pH - pHinp0) / dTempC
'        Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTempCinp = (pH - pHout0) / dTempC
                dfCO2outdTempCinp = (fCO2 - fCO2out0) / dTempC
                dpCO2outdTempCinp = (pCO2 - pCO2out0) / dTempC
' **************
'Increase Pdbarinp by dPdbar
```

```
'        Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp + dPdbar
                GOSUB GetConstantsforCase5Partials:
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase5Partials:
                dTAdPdbarinp = (TA - TA0) / dPdbar
                dpHinpdPdbarinp = (pH - pHinp0) / dPdbar
'        Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdPdbarinp = (pH - pHout0) / dPdbar
                dfCO2outdPdbarinp = (fCO2 - fCO2out0) / dPdbar
                dpCO2outdPdbarinp = (pCO2 - pCO2out0) / dPdbar
' **************
'Increase K0 by pcdK0 % at input conditions only
'        Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
                K0 = K0 * (1! + pcdK0 / 100!)
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase5Partials:
                dpHinppcdK0 = (pH - pHinp0) / pcdK0
                dTApcdK0 = (TA - TA0) / pcdK0
'        Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK0 = (pH - pHout0) / pcdK0
                dfCO2outpcdK0 = (fCO2 - fCO2out0) / pcdK0
                dpCO2outpcdK0 = (pCO2 - pCO2out0) / pcdK0
' **************
'Increase K1 by pcdK1 % at input conditions only
'        Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
                K(1) = K(1) * (1! + pcdK1 / 100!): K1 = K(1)
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase5Partials:
                dTApcdK1 = (TA - TA0) / pcdK1
                dpHinppcdK1 = (pH - pHinp0) / pcdK1
'        Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK1 = (pH - pHout0) / pcdK1
                dfCO2outpcdK1 = (fCO2 - fCO2out0) / pcdK1
                dpCO2outpcdK1 = (pCO2 - pCO2out0) / pcdK1
' **************
'Increase K2 by pcdK2 % at input conditions only
'        Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase5Partials:
                K(2) = K(2) * (1! + pcdK2 / 100!): K2 = K(2)
                fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase5Partials:
                dpHinppcdK2 = (pH - pHinp0) / pcdK2
                dTApcdK2 = (TA - TA0) / pcdK2
'        Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK2 = (pH - pHout0) / pcdK2
                dfCO2outpcdK2 = (fCO2 - fCO2out0) / pcdK2
                dpCO2outpcdK2 = (pCO2 - pCO2out0) / pcdK2
'
'
' ******************************************
        AAA$ = "####.#  #.####           #.####  ####.# "
PrintPartialsForCase5:
        IF fORp$ = "f" THEN
                PRINT USING "                             TA    \    \
\    \   fCO2 "; pHScale$; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TA0 *
1000000!; pHinp0; pHout0; fCO2out0 * 1000000!
                PRINT USING "    1 umol/kg in TC        " + AAA$; dTAdTC;
dpHinpdTC / 1000000!; dpHoutdTC / 1000000!; dfCO2outdTC
                PRINT USING "    1 uatm in input fCO2   " + AAA$; dTAdfCO2;
dpHinpdfCO2 / 1000000!; dpHoutdfCO2 / 1000000!; dfCO2outdfCO2
        ELSEIF fORp$ = "p" THEN
                PRINT USING "                             TA    \    \
\    \   pCO2 "; pHScale$; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                   " + AAA$; TA0 *
1000000!; pHinp0; pHout0; pCO2out0 * 1000000!
                PRINT USING "    1 umol/kg in TC        " + AAA$; dTAdTC;
dpHinpdTC / 1000000!; dpHoutdTC / 1000000!; dpCO2outdTC
                PRINT USING "    1 uatm in input pCO2   " + AAA$; dTAdpCO2;
dpHinpdpCO2 / 1000000!; dpHoutdpCO2 / 1000000!; dpCO2outdpCO2
        END IF
        IF WhichKs% <> 8 THEN
                PRINT USING "    1 in salinity          " + AAA$; dTAdSal *
1000000!; dpHinpdSal; dpHoutdSal; dfCO2outdSal * 1000000!
        END IF
        PRINT USING "    1 deg C in input T     " + AAA$; dTAdTempCinp *
1000000!; dpHinpdTempCinp; dpHoutdTempCinp; dfCO2outdTempCinp * 1000000!
        PRINT USING "    100 dbar in input P    " + AAA$; dTAdPdbarinp *
1000000! * 100!; dpHinpdPdbarinp * 100!; dpHoutdPdbarinp * 100!;
dfCO2outdPdbarinp * 1000000! * 100!
        PRINT USING "    1% K0 at input T, P    " + AAA$; dTApcdK0 * 1000000!;
dpHinppcdK0; dpHoutpcdK0; dfCO2outpcdK0 * 1000000!
        PRINT USING "    1% K1 at input T, P    " + AAA$; dTApcdK1 * 1000000!;
dpHinppcdK1; dpHoutpcdK1; dfCO2outpcdK1 * 1000000!
        PRINT USING "    1% K2 at input T, P    " + AAA$; dTApcdK2 * 1000000!;
dpHinppcdK2; dpHoutpcdK2; dfCO2outpcdK2 * 1000000!
'
'
        TA = TA0: ' to pass back the value that came in
EXIT SUB
'****************************************************************************
GetConstantsforCase5Partials:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
RETURN
CalculateStuffForCase5Partials:
        CALL CalculatepHfromTCfCO2(TC, fCO2, K0, K1, K2, pH)
        IF pH = -999! THEN
                TCfCO2Flag% = 1
                TA = TA0: ' to pass back the value that came in
                EXIT SUB
        '       this means that the TC, fCO2 combination becomes
        '       physically unrealizable during calculations in this sub
        END IF
        CALL CalculateTAfromTCpH(TC, pH, K(), T(), TA)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
RETURN
END SUB
SUB Case6Partials (pHScale$, WhichKs%, WhoseKSO4%, fORp$, Sal, K(), T(),
TempCinp, TempCout, Pdbarinp, Pdbarout, pHinp, fCO2inp, pCO2inp, TA, TC, pHout,
fCO2out, pCO2out)
' SUB Case6Partials, version 01.04, 03-12-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, fORp$
' Inputs: Sal, K(), T(), TempCinp, TempCout, Pdbarinp, Pdbarout
' Inputs: pHinp, fCO2inp, pCO2inp, TA, TC, pHout, fCO2out, pCO2out
' Outputs: none
' This calculates and prints the partials for case 6: input pH, fCO2 or pCO2.
'
'
        TA0 = TA: TC0 = TC: Sal0 = Sal
        pHinp0 = pHinp: pHout0 = pHout
        fCO2inp0 = fCO2inp
        fCO2out0 = fCO2out: pCO2out0 = pCO2out
        CALL SetParametersForPartials(dTA, dTC, dpH, dfCO2, dSal, dTempC,
dPdbar, pcdK0, pcdK1, pcdK2)
' **************
'Increase pH by dpH (this is pH at input conditions)
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
                pH = pHinp0 + dpH: fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase6Partials:
                dTAdpH = (TA - TA0) / dpH
                dTCdpH = (TC - TC0) / dpH
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdpH = (pH - pHout0) / dpH
                dfCO2outdpH = (fCO2 - fCO2out0) / dpH
                dpCO2outdpH = (pCO2 - pCO2out0) / dpH
' **************
'Increase fCO2 by dfCO2 (this is fCO2 at input conditions)
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
'               this must be after call to constants for correct FugFac
                dpCO2 = dfCO2 / FugFac
                pH = pHinp0: fCO2 = fCO2inp0 + dfCO2
                GOSUB CalculateStuffForCase6Partials:
                dTAdfCO2 = (TA - TA0) / dfCO2
                dTAdpCO2 = (TA - TA0) / dpCO2
                dTCdfCO2 = (TC - TC0) / dfCO2
                dTCdpCO2 = (TC - TC0) / dpCO2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdfCO2 = (pH - pHout0) / dfCO2
                dpHoutdpCO2 = (pH - pHout0) / dpCO2
                dfCO2outdfCO2 = (fCO2 - fCO2out0) / dfCO2
                dpCO2outdpCO2 = (pCO2 - pCO2out0) / dpCO2
' **************
'Increase Sal by dSal
        Sal = Sal0 + dSal
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
                pH = pHinp0: fCO2 = fCO2inp0
'               since conversion of pCO2 to fCO2 depends on Sal, Temp:
                IF fORp$ = "p" THEN fCO2 = pCO2inp * FugFac
                GOSUB CalculateStuffForCase6Partials:
                dTCdSal = (TC - TC0) / dSal
                dTAdSal = (TA - TA0) / dSal
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdSal = (pH - pHout0) / dSal
                dfCO2outdSal = (fCO2 - fCO2out0) / dSal
                dpCO2outdSal = (pCO2 - pCO2out0) / dSal
        Sal = Sal0
' **************
'Increase TempCinp by dTempC
'       Do at Tinp, Pinp
                TempC = TempCinp + dTempC: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
                pH = pHinp0: fCO2 = fCO2inp0
'               since conversion of pCO2 to fCO2 depends on Sal, Temp:
                IF fORp$ = "p" THEN fCO2 = pCO2inp * FugFac
                GOSUB CalculateStuffForCase6Partials:
                dTAdTempCinp = (TA - TA0) / dTempC
                dTCdTempCinp = (TC - TC0) / dTempC
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdTempCinp = (pH - pHout0) / dTempC
                dfCO2outdTempCinp = (fCO2 - fCO2out0) / dTempC
                dpCO2outdTempCinp = (pCO2 - pCO2out0) / dTempC
' **************
'Increase Pdbarinp by dPdbar
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp + dPdbar
                GOSUB GetConstantsforCase6Partials:
```

```
                pH = pHinp0: fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase6Partials:
                dTAdPdbarinp = (TA - TA0) / dPdbar
                dTCdPdbarinp = (TC - TC0) / dPdbar
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutdPdbarinp = (pH - pHout0) / dPdbar
                dfCO2outdPdbarinp = (fCO2 - fCO2out0) / dPdbar
                dpCO2outdPdbarinp = (pCO2 - pCO2out0) / dPdbar
' **************
'Increase K0 by pcdK0 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
                K0 = K0 * (1! + pcdK0 / 100!)
                pH = pHinp0: fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase6Partials:
                dTApcdK0 = (TA - TA0) / pcdK0
                dTCpcdK0 = (TC - TC0) / pcdK0
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK0 = (pH - pHout0) / pcdK0
                dfCO2outpcdK0 = (fCO2 - fCO2out0) / pcdK0
                dpCO2outpcdK0 = (pCO2 - pCO2out0) / pcdK0
' **************
'Increase K1 by pcdK1 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
                K(1) = K(1) * (1! + pcdK1 / 100!): K1 = K(1)
                pH = pHinp0: fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase6Partials:
                dTApcdK1 = (TA - TA0) / pcdK1
                dTCpcdK1 = (TC - TC0) / pcdK1
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK1 = (pH - pHout0) / pcdK1
                dfCO2outpcdK1 = (fCO2 - fCO2out0) / pcdK1
                dpCO2outpcdK1 = (pCO2 - pCO2out0) / pcdK1
' **************
'Increase K2 by pcdK2 % at input conditions only
'       Do at Tinp, Pinp
                TempC = TempCinp: Pdbar = Pdbarinp
                GOSUB GetConstantsforCase6Partials:
                K(2) = K(2) * (1! + pcdK2 / 100!): K2 = K(2)
                pH = pHinp0: fCO2 = fCO2inp0
                GOSUB CalculateStuffForCase6Partials:
                dTApcdK2 = (TA - TA0) / pcdK2
                dTCpcdK2 = (TC - TC0) / pcdK2
'       Do at Tout, Pout
                TempC = TempCout: Pdbar = Pdbarout
                CALL FindpHfCO2fromTATC(pHScale$, WhichKs%, WhoseKSO4%, TA, TC,
Sal, K(), T(), TempC, Pdbar, pH, fCO2, pCO2)
                dpHoutpcdK2 = (pH - pHout0) / pcdK2
                dfCO2outpcdK2 = (fCO2 - fCO2out0) / pcdK2
                dpCO2outpcdK2 = (pCO2 - pCO2out0) / pcdK2
'
'
' ******************************************
        AAA$ = "####.#  ####.#           #.####  ####.# "
PrintPartialsForCase6:
        IF fORp$ = "f" THEN
                PRINT USING "                             TA      TC
\    \   fCO2 "; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                  " + AAA$; TA0 *
1000000!; TC0 * 1000000!; pHout0; fCO2out0 * 1000000!
                PRINT USING "    .001 in input pH       " + AAA$; dTAdpH *
1000000! / 1000!; dTCdpH * 1000000! / 1000!; dpHoutdpH / 1000!; dfCO2outdpH *
1000000! / 1000!
                PRINT USING "    1 uatm in input fCO2   " + AAA$; dTAdfCO2;
dTCdfCO2; dpHoutdfCO2 / 1000000!; dfCO2outdfCO2
        ELSEIF fORp$ = "p" THEN
                PRINT USING "                             TA      TC
\    \   pCO2 "; pHScale$
                PRINT "    change per             ------  ------
------  ------ "
                PRINT USING "     \ /                  " + AAA$; TA0 *
1000000!; TC0 * 1000000!; pHout0; pCO2out0 * 1000000!
                PRINT USING "    .001 in input pH       " + AAA$; dTAdpH *
1000000! / 1000!; dTCdpH * 1000000! / 1000!; dpHoutdpH / 1000!; dpCO2outdpH *
1000000! / 1000!
                PRINT USING "    1 uatm in input pCO2   " + AAA$; dTAdpCO2;
dTCdpCO2; dpHoutdpCO2 / 1000000!; dpCO2outdpCO2
        END IF
        IF WhichKs% <> 8 THEN
                PRINT USING "    1 in salinity          " + AAA$; dTAdSal *
1000000!; dTCdSal * 1000000!; dpHoutdSal; dfCO2outdSal * 1000000!
        END IF
        PRINT USING "    1 deg C in input T     " + AAA$; dTAdTempCinp *
1000000!; dTCdTempCinp * 1000000!; dpHoutdTempCinp; dfCO2outdTempCinp * 1000000!
        PRINT USING "    100 dbar in input P    " + AAA$; dTAdPdbarinp *
1000000! * 100!; dTCdPdbarinp * 1000000! * 100!; dpHoutdPdbarinp * 100!;
dfCO2outdPdbarinp * 1000000! * 100!
        PRINT USING "    1% K0 at input T       " + AAA$; dTApcdK0 * 1000000!;
dTCpcdK0 * 1000000!; dpHoutpcdK0; dfCO2outpcdK0 * 1000000!
        PRINT USING "    1% K1 at input T, P    " + AAA$; dTApcdK1 * 1000000!;
dTCpcdK1 * 1000000!; dpHoutpcdK1; dfCO2outpcdK1 * 1000000!
        PRINT USING "    1% K2 at input T, P    " + AAA$; dTApcdK2 * 1000000!;
dTCpcdK2 * 1000000!; dpHoutpcdK2; dfCO2outpcdK2 * 1000000!
'
'
        TA = TA0: TC = TC0: ' to pass back the values that came in
EXIT SUB
'***************************************************************************
GetConstantsforCase6Partials:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
RETURN
CalculateStuffForCase6Partials:
        CALL CalculateTCfrompHfCO2(pH, fCO2, K0, K1, K2, TC)
        CALL CalculateTAfromTCpH(TC, pH, K(), T(), TA)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
RETURN
END SUB
SUB CaSolubility (WhichKs%, Sal, TempC, Pdbar, TC, pH, K1, K2, OmegaCa, OmegaAr)
' SUB CaSolubility, version 01.05, 05-23-97, written by Ernie Lewis.
' Inputs: WhichKs%, Sal, TempC, Pdbar, TC, pH, K1, K2
' Outputs: OmegaCa, OmegaAr
' This calculates omega, the solubility ratio, for calcite and aragonite.
' This is defined by: Omega = [CO3--]*[Ca++] / Ksp,
'       where Ksp is the solubility product (either KCa or KAr).
'
'
'****************************************************************************
' These are from:
' Mucci, Alphonso, The solubility of calcite and aragonite in seawater
'       at various salinities, temperatures, and one atmosphere total
'       pressure, American Journal of Science 283:781-799, 1983.
' Ingle, S. E., Solubility of calcite in the ocean,
'       Marine Chemistry 3:301-319, 1975,
' Millero, Frank, The thermodynamics of the carbonate system in seawater,
'       Geochemica et Cosmochemica Acta 43:1651-1661, 1979.
' Ingle et al, The solubility of calcite in seawater at atmospheric pressure
'       and 35%o salinity, Marine Chemistry 1:295-307, 1973.
' Berner, R. A., The solubility of calcite and aragonite in seawater in
'       atmospheric pressure and 34.5%o salinity, American Journal of
'       Science 276:713-730, 1976.
' Takahashi et al, in GEOSECS Pacific Expedition, v. 3, 1982.
' Culberson, C. H. and Pytkowicz, R. M., Effect of pressure on carbonic acid,
'       boric acid, and the pH of seawater, Limnology and Oceanography
'       13:403-417, 1968.
'
'
'****************************************************************************
        RGasConstant = 83.1451: 'bar-cm3/(mol-K)
        TempK = TempC + 273.15
        RT = RGasConstant * TempK
        Pbar = Pdbar / 10!
        logTempK = LOG(TempK)
        sqrSal = SQR(Sal)
'       deltaVs are in cm3/mole
'       Kappas are in cm3/mole/bar
'       PROGRAMMER'S NOTE: all logs are log base e
'
'
'****************************************************************************
CalculateCa:
'       Riley, J. P. and Tongudai, M., Chemical Geology 2:263-269, 1967:
        Ca = .02128 / 40.087 * (Sal / 1.80655): ' in mol/kg-SW
'       this is .010285 * Sal / 35
'
```

```
'
CalciteSolubility:
'       Mucci, Alphonso, Amer. J. of Science 283:781-799, 1983.
        logKCa = -171.9065 - .077993 * TempK + 2839.319 / TempK
        logKCa = logKCa + 71.595 * logTempK / LOG(10!)
        logKCa = logKCa + (-.77712 + .0028426 * TempK + 178.34 / TempK) * sqrSal
        logKCa = logKCa - .07711 * Sal + .0041249 * sqrSal * Sal
'       sd fit = .01 (for Sal part, not part independent of Sal)
        KCa = 10! ^ (logKCa): ' this is in (mol/kg-SW)^2
'
'
AragoniteSolubility:
'       Mucci, Alphonso, Amer. J. of Science 283:781-799, 1983.
        logKAr = -171.945 - .077993 * TempK + 2903.293 / TempK
        logKAr = logKAr + 71.595 * logTempK / LOG(10!)
        logKAr = logKAr + (-.068393 + .0017276 * TempK + 88.135 / TempK) *
sqrSal
        logKAr = logKAr - .10018 * Sal + .0059415 * sqrSal * Sal
'       sd fit = .009 (for Sal part, not part independent of Sal)
        KAr = 10! ^ (logKAr): ' this is in (mol/kg-SW)^2
'
'
PressureCorrectionForCalcite:
'       Ingle, Marine Chemistry 3:301-319, 1975
'       same as in Millero, GCA 43:1651-1661, 1979, but Millero, GCA 1995
'       has typos (-.5304, -.3692, and 10^3 for Kappa factor)
        deltaVKCa = -48.76 + .5304 * TempC
        KappaKCa = (-11.76 + .3692 * TempC) / 1000!
        lnKCafac = (-deltaVKCa + .5 * KappaKCa * Pbar) * Pbar / RT
        KCa = KCa * EXP(lnKCafac)
'
'
PressureCorrectionForAragonite:
'       Millero, Geochemica et Cosmochemica Acta 43:1651-1661, 1979,
'       same as Millero, GCA 1995 except for typos (-.5304, -.3692,
'       and 10^3 for Kappa factor)
        deltaVKAr = deltaVKCa + 2.8
        KappaKAr = KappaKCa
        lnKArfac = (-deltaVKAr + .5 * KappaKAr * Pbar) * Pbar / RT
        KAr = KAr * EXP(lnKArfac)
'
'
'****************************************************************************
IF WhichKs% = 6 OR WhichKs% = 7 THEN
CalculateCaforGEOSECS:
'       Culkin, F, in Chemical Oceanography, ed. Riley and Skirrow, 1965:
'       (quoted in Takahashi et al, GEOSECS Pacific Expedition v. 3, 1982)
        Ca = .01026 * Sal / 35!
'       Culkin gives Ca = (.0213 / 40.078) * (Sal / 1.80655) in mol/kg-SW
'       which corresponds to Ca = .01030 * Sal / 35.
'
'
CalculateKCaforGEOSECS:
'       Ingle et al, Marine Chemistry 1:295-307, 1973 is referenced in
'       (quoted in Takahashi et al, GEOSECS Pacific Expedition v. 3, 1982
'       but the fit is actually from Ingle, Marine Chemistry 3:301-319, 1975)
```

```
        KCa = .0000001 * (-34.452 - 39.866 * Sal ^ (1 / 3) + 110.21 * LOG(Sal) /
LOG(10!) - 7.5752E-06 * TempK * TempK)
'       this is in (mol/kg-SW)^2
'
'
CalculateKArforGEOSECS:
'       Berner, R. A., American Journal of Science 276:713-730, 1976:
'       (quoted in Takahashi et al, GEOSECS Pacific Expedition v. 3, 1982)
        KAr = 1.45 * KCa: ' this is in (mol/kg-SW)^2
'       Berner (p. 722) states that he uses 1.48.
'       It appears that 1.45 was used in the GEOSECS calculations
'
'
CalculatePressureEffectsOnKCaKArGEOSECS:
'       Culberson and Pytkowicz, Limnology and Oceanography 13:403-417, 1968
'       (quoted in Takahashi et al, GEOSECS Pacific Expedition v. 3, 1982
'       but their paper is not even on this topic).
'       The fits appears to be new in the GEOSECS report.
'       I can't find them anywhere else.
        KCa = KCa * EXP((36! - .2 * TempC) * Pbar / RT)
        KAr = KAr * EXP((33.3 - .22 * TempC) * Pbar / RT)
END IF
'
'
'****************************************************************************
CalculateOmegasHere:
        H = 10! ^ (-pH)
        CO3 = TC * K1 * K2 / (K1 * H + H * H + K1 * K2)
        OmegaCa = CO3 * Ca / KCa: 'dimensionless
        OmegaAr = CO3 * Ca / KAr: 'dimensionless
END SUB
SUB ChoosefORp (fORp$)
' SUB ChoosefORp$, version 02.02, 10-10-97, written by Ernie Lewis.
' Inputs: fORp$
' Outputs: fORp$
' This allows a choice of fCO2 or pCO2 as the variable to use.
' They are related by pCO2 = fCO2 / FugFac, where
'       FugFac = EXP((B+2*Delta)*P/RT) accounts for non-ideality of CO2
'       (this converts fugacity to partial pressure and assumes xCO2 << 1),
'
'
        IF fORp$ = "f" THEN fORpDefault% = 1
        IF fORp$ = "p" THEN fORpDefault% = 2
TopOfChoosefORp:
        CLS
        PRINT USING "    Choose one of the following (#): "; fORpDefault%
        PRINT
        PRINT "         1) fCO2, the fugacity of CO2 "
        PRINT "         2) pCO2, the partial pressure of CO2 "
        PRINT
        PRINT "    Enter <I> for more information on fCO2 and pCO2. "
'
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 38: INPUT "", fORpChoice$
        IF fORpChoice$ = "I" OR fORpChoice$ = "i" THEN
                Info$ = "fCO2pCO2": CALL AboutCO2SYS(Info$)
                GOTO TopOfChoosefORp:
        ELSE
                fORp% = VAL(fORpChoice$)
        END IF
        SELECT CASE fORp%
                CASE 0
                        ' then there is no change in fORp$
                        ' and the current value is used
                CASE 1
                        fORp$ = "f"
                CASE 2
                        fORp$ = "p"
                CASE ELSE
                        BEEP
                        GOTO TopOfChoosefORp:
        END SELECT
END SUB
SUB ChooseInputMode (Batch$)
' SUB ChooseInputMode, version 01.02, 05-03-97, written by Ernie Lewis.
' Inputs: Batch$
' Outputs: Batch$
' This allows the user to choose between single-input mode (Batch$ = "NO")
'       and batch-input mode where the data are read in from a file
'       (Batch$ = "YES").
'
'
        IF Batch$ = "NO" THEN BatchDefault% = 1
        IF Batch$ = "YES" THEN BatchDefault% = 2
'
'
TopOfChooseInputMode:
        CLS
        PRINT USING "    Choose one of the following (#): "; BatchDefault%
        PRINT
        PRINT "         1) single-input mode "
        PRINT "         2) batch-input mode "
        PRINT
        PRINT "    Enter <I> for more information on batch-input mode. "
'
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 38: INPUT "", BatchModeChoice$
        IF BatchModeChoice$ = "I" OR BatchModeChoice$ = "i" THEN
                Info$ = "Batch": CALL AboutCO2SYS(Info$)
                GOTO TopOfChooseInputMode:
        ELSE
                Batch% = VAL(BatchModeChoice$)
        END IF
        SELECT CASE Batch%
                CASE 0
                        ' then there is no change in Batch$
                        ' and the current value is used
                CASE 1
                        Batch$ = "NO"
                CASE 2
                        Batch$ = "YES"
                CASE ELSE
```

```
                        BEEP
                        GOTO TopOfChooseInputMode:
        END SELECT
END SUB
SUB ChoosepHScale (pHScale$)
' SUB ChoosepHScale, version 01.03, 03-03-97, written by Ernie Lewis.
' Inputs: pHScale$
' Outputs: pHScale$
' This allows the user to choose the pH scale inputs will be made on, and
'       on which the results will be given.
'
'
        SELECT CASE pHScale$
                CASE "pHtot"
                        pHScaleDefault% = 1
                CASE "pHsws"
                        pHScaleDefault% = 2
                CASE "pHfree"
                        pHScaleDefault% = 3
                CASE "pHNBS"
                        pHScaleDefault% = 4
                CASE "pH"
                        ' this is for WhichKs% = 8 (freshwater choice).
                        ' The scales are all the same.
                        ' This is not a choice here but is automatically
                        ' set when WhichKs% = 8 is chosen.
                        EXIT SUB
        END SELECT
'
'
TopOfChoosepHScale:
        CLS
        PRINT USING "    Choose the pH scale here (#): "; pHScaleDefault%
        PRINT
        PRINT "         1) Total scale (mol/kg-SW) "
        PRINT "         2) Seawater scale (mol/kg-SW) "
        PRINT "         3) Free scale (mol/kg-SW) "
        PRINT "         4) NBS scale (mol/kg-H2O) "
        PRINT
        PRINT "    Enter <I> for information on the various pH scales. "
'
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 35: INPUT "", WhichpHScale$
        IF WhichpHScale$ = "I" OR WhichpHScale$ = "i" THEN
                Info$ = "pHScales": CALL AboutCO2SYS(Info$)
                GOTO TopOfChoosepHScale:
        ELSE
                WhichpHScale% = VAL(WhichpHScale$)
        END IF
        SELECT CASE WhichpHScale%
                CASE 0
                        ' then there is no change in pHScale$
                        ' and the current value is used
                CASE 1
                        pHScale$ = "pHtot"
                CASE 2
```

```
                        pHScale$ = "pHsws"
                CASE 3
                        pHScale$ = "pHfree"
                CASE 4
                        pHScale$ = "pHNBS"
                CASE ELSE
                        BEEP
                        GOTO TopOfChoosepHScale:
        END SELECT
END SUB
SUB ChooseWhichCase (ICase%, fORp$)
' SUB ChooseWhichCase, version 01.03, 03-12-97, written by Ernie Lewis.
' Inputs: ICase%, fORp$
' Outputs: ICase%
' This prints out choices and takes as input the desired one.
'
'
        ICaseDefault% = ICase%
TopOfChooseWhichCase:
        CLS
        PRINT USING "    Choose one of the following (#): "; ICaseDefault%
        SELECT CASE fORp$
                CASE "f"
                        PRINT
                        PRINT "             GIVEN             CALCULATE "
                        PRINT "             -----             --------- "
                        PRINT "          1) TA, TC            pH, fCO2 "
                        PRINT "          2) TA, pH            TC, fCO2 "
                        PRINT "          3) TA, fCO2          TC, pH "
                        PRINT "          4) TC, pH            TA, fCO2 "
                        PRINT "          5) TC, fCO2          TA, pH "
                        PRINT "          6) pH, fCO2          TA, TC "
                CASE "p"
                        PRINT
                        PRINT "          1) TA, TC            pH, pCO2 "
                        PRINT "          2) TA, pH            TC, pCO2 "
                        PRINT "          3) TA, pCO2          TC, pH "
                        PRINT "          4) TC, pH            TA, pCO2 "
                        PRINT "          5) TC, pCO2          TA, pH "
                        PRINT "          6) pH, pCO2          TA, TC "
        END SELECT
'
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 38: INPUT "", ICase%
        SELECT CASE ICase%
                CASE 0
                        ICase% = ICaseDefault%
                CASE 1, 2, 3, 4, 5, 6
                        ' nothing is to be done here, the case is chosen
                CASE ELSE
                        BEEP
                        GOTO TopOfChooseWhichCase:
        END SELECT
END SUB
SUB ChooseWhichKs (WhichKs%, pHScale$)
' Sub ChooseWhichKs, version 02.03, 05-02-97, written by Ernie Lewis.
' Inputs: WhichKs%, pHScale$
' Outputs: WhichKs%, pHScale$
' This allows the user a choice of which set of constants to use.
'
'
        WhichKsDefault% = WhichKs%
TopOfChooseWhichKs:
        CLS
        PRINT USING "    Choose one of the following values for the constants
(#): "; WhichKsDefault%
        PRINT
        PRINT "         1) K1, K2 from Roy, et al, 1993 "
        'PRINT "                 2s PRECISION about 2% in K1 and 1.5% in K2. "
        PRINT "         2) K1, K2 from Goyet and Poisson, 1989 "
        'PRINT "                 2s PRECISION about 2.5% in K1 and 4.5% in K2. "
        PRINT "         3) K1, K2 from Hansson, 1973 refit by Dickson and
Millero, 1987 "
        'PRINT "                 2s PRECISION about 3% in K1 and 4% in K2. "
        PRINT "         4) K1, K2 from Mehrbach et al, 1973 refit by Dickson and
Millero, 1987 "
        'PRINT "                 2s PRECISION about 2.5% in K1 and 4.5% in K2. "
        PRINT "         5) K1, K2 from Hansson and Mehrbach refit by Dickson and
Millero, 1987 "
        'PRINT "                 2s PRECISION about 4% in K1 and 6% in K2. "
        PRINT "         6) GEOSECS constants (NBS scale); K1, K2 from Mehrbach
et al, 1973 "
        'PRINT "                 2s PRECISION about 1.2% in K1 and 2.0% in K2. "
        PRINT "         7) Constants from Peng et al (NBS scale); K1, K2 from
Mehrbach et al "
        'PRINT "                 2s PRECISION about 1.2% in K1 and 2.0% in K2. "
        PRINT "         8) Salinity = 0 (freshwater); K1, K2 from Millero 1979 "
        'PRINT "                 2s PRECISION about .5% in K1 and .7% in K2. "
        PRINT
        PRINT "    For 6) or 7), the pH scale is set to NBS. This can be changed
later. "
        PRINT "    In each case the constants are converted to the chosen pH
scale. "
        PRINT
        PRINT "    For information on the constants used in general, or cases 1-
5, enter <C>, "
        PRINT "    For information on GEOSECS, enter <G>, "
        PRINT "    For information on the Peng case, enter <P>, "
        PRINT "    For information on the freshwater case, enter <F>. "
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 63: INPUT "", WhichKs$
        SELECT CASE WhichKs$
                CASE "C", "c"
                        Info$ = "Constants": CALL AboutCO2SYS(Info$)
                CASE "G", "g"
                        Info$ = "GEOSECS": CALL AboutCO2SYS(Info$)
                CASE "P", "p"
                        Info$ = "Peng": CALL AboutCO2SYS(Info$)
                CASE "F", "f"
                        Info$ = "Freshwater": CALL AboutCO2SYS(Info$)
                CASE ""
                       WhichKs% = WhichKsDefault%
                       EXIT SUB
                CASE "1", "2", "3", "4", "5"
                        ' nothing is to be done here, the case is chosen
                        WhichKs% = VAL(WhichKs$)
                        EXIT SUB
                CASE "6", "7"
                        pHScale$ = "pHNBS"
                        ' default pH scale for these cases is NBS, this can
                        ' be overridden later
                        WhichKs% = VAL(WhichKs$)
                        EXIT SUB
                CASE "8"
                        WhichKs% = VAL(WhichKs$)
                        pHScale$ = "pH"
                        EXIT SUB
        END SELECT
        GOTO TopOfChooseWhichKs:
END SUB
SUB ChooseWhoseKSO4 (WhoseKSO4%)
' SUB ChooseWhoseKSO4, version 01.02, 01-02-97, written by Ernie Lewis.
' Inputs: WhoseKSO4%
' Outputs: WhoseKSO4%
' This allows a user choice of whose KSO4 constant to use.
'
'
        WhoseKSO4Default% = WhoseKSO4%
TopOfChooseKSO4:
        CLS
        PRINT USING "    Choose one of the following values for KSO4 (#):   ";
WhoseKSO4Default%
        PRINT
        PRINT "         1) Dickson "
        PRINT "         2) Khoo et al "
        PRINT
        PRINT "    Enter <I> for more information on KSO4. "
'
'
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 1, 54: INPUT "", WhoseKSO4Choice$
        IF WhoseKSO4Choice$ = "I" OR WhoseKSO4Choice$ = "i" THEN
                Info$ = "KSO4": CALL AboutCO2SYS(Info$)
                GOTO TopOfChooseKSO4:
        ELSE
                WhoseKSO4% = VAL(WhoseKSO4Choice$)
        END IF
        SELECT CASE WhoseKSO4%
                CASE 0
                        WhoseKSO4% = WhoseKSO4Default%
                CASE 1, 2
                        ' nothing is to be done here, the case is chosen
                CASE ELSE
                        BEEP
                        GOTO TopOfChooseKSO4:
        END SELECT
END SUB
SUB Constants (pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0, K(), T(),
fH, FugFac, VPFac)
```

```
' SUB Constants, version 04.01, 10-13-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar
' Outputs: K0, K(), T(), fH, FugFac, VPFac
' This finds the constants of the CO2 system in seawater or freshwater,
' corrects them for pressure, and reports them on the chosen pH scale.
' The process is as follows: the constants (except KS, KF which stay on the
' free scale - these are only corrected for pressure) are
'       1) evaluated as they are given in the literature
'       2) converted to the SWS scale in mol/kg-SW or to the NBS scale
'       3) corrected for pressure
'       4) converted to the SWS pH scale in mol/kg-SW
'       5) converted to the chosen pH scale
'
'
'*******************************
'       PROGRAMMER'S NOTE: all logs are log base e
'       PROGRAMMER'S NOTE: all constants are converted to the pH scale
'               pHScale$ (the chosen one) in units of mol/kg-SW
'               except KS and KF are on the free scale
'               and KW is in units of (mol/kg-SW)^2
'
'
'*******************************
        RGasConstant = 83.1451: 'bar-cm3/(mol-K): ' = 8.31451 N-m/(mol-K)
        TempK = TempC + 273.15
        RT = RGasConstant * TempK
        sqrSal = SQR(Sal)
        logTempK = LOG(TempK)
        Pbar = Pdbar / 10!
'       deltaVs are in cm3/mole
'       Kappas are in cm3/mole/bar
'
'
'****************************************************************************
CalculateTB:
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5
                ' Uppstrom, L., Deep-Sea Research 21:161-162, 1974:
                TB = (.000232 / 10.811) * (Sal / 1.80655): ' in mol/kg-SW
                ' this is .000416 * Sal / 35. = .0000119 * Sal
        CASE 6, 7
                ' Culkin, F., in Chemical Oceanography,
                ' ed. Riley and Skirrow, 1965:
                ' GEOSECS references this, but this value is not explicitly
                ' given here
                TB = .0004106 * Sal / 35!: ' in mol/kg-SW
                ' this is .00001173 * Sal
                ' this is about 1% lower than Uppstrom's value
        CASE 8
                TB = 0!
        END SELECT
'
'
'*******************************
CalculateTF:
        ' Riley, J. P., Deep-Sea Research 12:219-220, 1965:
        TF = (.000067 / 18.998) * (Sal / 1.80655): ' in mol/kg-SW
        ' this is .000068 * Sal / 35. = .00000195 * Sal
'
'
'*******************************
CalculateTS:
        ' Morris, A. W., and Riley, J. P., Deep-Sea Research 13:699-705, 1966:
        TS = (.14 / 96.062) * (Sal / 1.80655): ' in mol/kg-SW
        ' this is .02824 * Sal / 35. = .0008067 * Sal
'
'
'*******************************
MakeTMatrix:
        T(1) = TB
        T(2) = TF
        T(3) = TS
'       T(4) = TP
'       T(5) = TSi
'       These last two were set earlier.
'
'
'****************************************************************************
CalculateK0:
        ' Weiss, R. F., Marine Chemistry 2:203-215, 1974.
        TTT = TempK / 100!
        lnK0 = -60.2409 + 93.4517 / TTT + 23.3585 * LOG(TTT)
        lnK0 = lnK0 + Sal * (.023517 - .023656 * TTT + .0047036 * TTT * TTT)
        K0 = EXP(lnK0): ' this is in mol/kg-SW/atm
'
'
'****************************************************************************
CalculateIonS:
'       This is from the DOE handbook, Chapter 5, p. 13/22, eq. 7.2.4:
        IonS = 19.924 * Sal / (1000! - 1.005 * Sal)
'
'
'****************************************************************************
CalculateKS:
        SELECT CASE WhoseKSO4%
        CASE 1: ' Dickson's value
'               Dickson, A. G., J. Chemical Thermodynamics, 22:113-127, 1990
'               The goodness of fit is .021.
'               It was given in mol/kg-H2O. I convert it to mol/kg-SW.
'               TYPO!!!!!! on p. 121: the constant e9 should be e8.
'
'       This is from eqs 22 and 23 on p. 123, and Table 4 on p 121:
        lnKS = -4276.1 / TempK + 141.328 - 23.093 * logTempK
        lnKS = lnKS + (-13856! / TempK + 324.57 - 47.986 * logTempK) * SQR(IonS)
        lnKS = lnKS + (35474! / TempK - 771.54 + 114.723 * logTempK) * IonS
        lnKS = lnKS + (-2698! / TempK) * SQR(IonS) * IonS
        lnKS = lnKS + (1776! / TempK) * IonS * IonS
        KS = EXP(lnKS): ' this is on the free pH scale in mol/kg-H2O
        KS = KS * (1! - .001005 * Sal): ' convert to mol/kg-SW
'
'
        CASE 2
'               Khoo, et al, Analytical Chemistry, 49(1):29-34, 1977
'               KS was found by titrations with a hydrogen electrode
'               of artificial seawater containing sulfate (but without F)
'               at 3 salinities from 20 to 45 and artificial seawater NOT
'               containing sulfate (nor F) at 16 salinities from 15 to 45,
'               both at temperatures from 5 to 40 deg C.
'               KS is on the Free pH scale (inherently so).
'               It was given in mol/kg-H2O. I convert it to mol/kg-SW.
'               He finds log(beta) which = my pKS;
'               his beta is an association constant.
'               The rms error is .0021 in pKS, or about .5% in KS.
'
'               This is equation 20 on p. 33:
        pKS = 647.59 / TempK - 6.3451 + .019085 * TempK - .5208 * SQR(IonS)
        KS = 10! ^ (-pKS): ' this is on the free pH scale in mol/kg-H2O
        KS = KS * (1! - .001005 * Sal): ' convert to mol/kg-SW
        END SELECT
'
'
'****************************************************************************
CalculateKF:
        ' Dickson, A. G. and Riley, J. P., Marine Chemistry 7:89-99, 1979:
        lnKF = 1590.2 / TempK - 12.641 + 1.525 * SQR(IonS)
        KF = EXP(lnKF): ' this is on the free pH scale in mol/kg-H2O
        KF = KF * (1! - .001005 * Sal): ' convert to mol/kg-SW
'
'
'****************************************************************************
CalculatepHScaleConversionFactors:
'       These are NOT pressure-corrected.
        SWStoTOT = (1! + TS / KS) / (1! + TS / KS + TF / KF)
        FREEtoTOT = 1! + TS / KS
'
'
'
CalculatefH:
'       Use GEOSECS's value for cases 1,2,3,4,5 (and 6) to convert pH scales.
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5, 6
                ' Takahashi et al, Chapter 3 in GEOSECS Pacific Expedition,
                ' v. 3, 1982 (p. 80):
                fH = 1.2948 - .002036 * TempK
                fH = fH + (.0004607 - 1.475E-06 * TempK) * Sal * Sal
        CASE 7
                ' Peng et al, Tellus 39B:439-458, 1987:
                ' They reference the GEOSECS report, but round the value
                ' given there off so that it is about .008 (1%) lower. It
                ' doesn't agree with the check value they give on p. 456.
                fH = 1.29 - .00204 * TempK
                fH = fH + (.00046 - 1.48E-06 * TempK) * Sal * Sal
        CASE 8
                fH = 1!: ' this shouldn't occur in the program for this case
        END SELECT
'
'
'****************************************************************************
CalculateKB:
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5
```

```
                ' Dickson, A. G., Deep-Sea Research 37:755-766, 1990:
                lnKBtop = -8966.9 - 2890.53 * sqrSal - 77.942 * Sal
                lnKBtop = lnKBtop + 1.728 * sqrSal * Sal - .0996 * Sal * Sal
                lnKB = lnKBtop / TempK
                lnKB = lnKB + 148.0248 + 137.1942 * sqrSal + 1.62142 * Sal
                lnKB = lnKB + (-24.4344 - 25.085 * sqrSal - .2474 * Sal) *
logTempK
                lnKB = lnKB + .053105 * sqrSal * TempK
                KB = EXP(lnKB): ' this is on the total pH scale in mol/kg-SW
                KB = KB / SWStoTOT: ' convert to SWS pH scale
'
'
        CASE 6, 7
                ' This is for GEOSECS and Peng et al.
                ' Lyman, John, UCLA Thesis, 1957
                ' fit by Li et al, JGR 74:5507-5525, 1969:
                logKB = -9.26 + .00886 * Sal + .01 * TempC
                KB = 10! ^ (logKB): ' this is on the NBS scale
                KB = KB / fH: ' convert to the SWS scale
'
'
        CASE 8
                KB = 0!
        END SELECT
'
'
'****************************************************************************
CalculateKW:
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5
                ' Millero, Geochemica et Cosmochemica Acta 59:661-677, 1995.
                ' his check value of 1.6 umol/kg-SW should be 6.2
                lnKW = 148.9802 - 13847.26 / TempK - 23.6521 * logTempK
                lnKW = lnKW + (-5.977 + 118.67 / TempK + 1.0495 * logTempK) *
sqrSal
                lnKW = lnKW - .01615 * Sal
                KW = EXP(lnKW): ' this is on the SWS pH scale in (mol/kg-SW)^2
'
'
        CASE 6
                KW = 0!: ' GEOSECS doesn't include OH effects
'
'
        CASE 7
                ' Millero, Geochemica et Cosmochemica Acta 43:1651-1661, 1979
                lnKW = 148.9802 - 13847.26 / TempK - 23.6521 * logTempK
                lnKW = lnKW + (-79.2447 + 3298.72 / TempK + 12.0408 * logTempK)
* sqrSal
                lnKW = lnKW - .019813 * Sal
                KW = EXP(lnKW): ' this is on the SWS pH scale
'
'
        CASE 8
                ' Millero, Geochemica et Cosmochemica Acta 43:1651-1661, 1979
                ' refit data of Harned and Owen, The Physical Chemistry of
                ' Electrolyte Solutions, 1958
                lnKW = 148.9802 - 13847.26 / TempK - 23.6521 * logTempK
                KW = EXP(lnKW)
        END SELECT
'
'
'***************************************************************************
CalculateKP1KP2KP3KSi:
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5
                ' Yao and Millero, Aquatic Geochemistry 1:53-88, 1995
                ' KP1, KP2, KP3 are on the SWS pH scale in mol/kg-SW.
                ' KSi was given on the SWS pH scale in molal units.
'
                lnKP1 = -4576.752 / TempK + 115.54 - 18.453 * logTempK
                lnKP1 = lnKP1 + (-106.736 / TempK + .69171) * sqrSal
                lnKP1 = lnKP1 + (-.65643 / TempK - .01844) * Sal
                KP1 = EXP(lnKP1)
'
'
                lnKP2 = -8814.715 / TempK + 172.1033 - 27.927 * logTempK
                lnKP2 = lnKP2 + (-160.34 / TempK + 1.3566) * sqrSal
                lnKP2 = lnKP2 + (.37335 / TempK - .05778) * Sal
                KP2 = EXP(lnKP2)
'
'
                lnKP3 = -3070.75 / TempK - 18.126
                lnKP3 = lnKP3 + (17.27039 / TempK + 2.81197) * sqrSal
                lnKP3 = lnKP3 + (-44.99486 / TempK - .09984) * Sal
                KP3 = EXP(lnKP3)
'
'
                lnKSi = -8904.2 / TempK + 117.4 - 19.334 * logTempK
                lnKSi = lnKSi + (-458.79 / TempK + 3.5913) * SQR(IonS)
                lnKSi = lnKSi + (188.74 / TempK - 1.5998) * IonS
                lnKSi = lnKSi + (-12.1652 / TempK + .07871) * IonS * IonS
                KSi = EXP(lnKSi): ' this is on the SWS pH scale in mol/kg-H2O
                KSi = KSi * (1! - .001005 * Sal): ' convert to mol/kg-SW
'
'
        CASE 7
                KP1 = .02:
                ' Peng et al don't include the contribution from this term,
                ' but it is so small it doesn't contribute. It needs to be
                ' kept so that the routines work ok.
                ' KP2, KP3 from Kester, D. R., and Pytkowicz, R. M.,
                ' Limnology and Oceanography 12:243-252, 1967:
                ' these are only for sals 33 to 36 and are on the NBS scale
                KP2 = EXP(-9.039 - 1450! / TempK): ' this is on the NBS scale
                KP2 = KP2 / fH: ' convert to SWS scale
                KP3 = EXP(4.466 - 7276 / TempK): ' this is on the NBS scale
                KP3 = KP3 / fH: ' convert to SWS scale
                ' Sillen, Martell, and Bjerrum,  Stability constants of metal-
ion complexes,
                ' The Chemical Society (London), Special Publ. 17:751, 1964:
                KSi = 4E-10:   ' this is on the NBS scale
                KSi = KSi / fH: ' convert to SWS scale
        CASE 6, 8
                KP1 = 0!
                KP2 = 0!
                KP3 = 0!
                KSi = 0!
                ' Neither the GEOSECS choice nor the freshwater choice
                ' include contributions from phosphate or silicate.
        END SELECT
'
'
'****************************************************************************
CalculateK1K2:
        SELECT CASE WhichKs%
'************************************************
        CASE 1: ' ROY et al, Marine Chemistry, 44:249-267, 1993
'               (see also: Erratum, Marine Chemistry 45:337, 1994
'               and Erratum, Marine Chemistry 52:183, 1996)
'               !!! Typo: in the abstract on p. 249: in the eq. for lnK1* the
'                       last term should have S raised to the power 1.5.
'               They claim standard deviations (p. 254) of the fits as
'               .0048 for lnK1 (.5% in K1) and .007 in lnK2 (.7% in K2).
'               They also claim (p. 258) 2s precisions of .004 in pK1 and
'               .006 in pK2. These are consistent, but Andrew Dickson
'               (personal communication) obtained an rms deviation of about
'               .004 in pK1 and .003 in pK2. This would be a 2s precision
'               of about 2% in K1 and 1.5% in K2.
'
'
'       This is eq. 29 on p. 254 and what they use in their abstract:
        lnK1 = 2.83655 - 2307.1266# / TempK - 1.5529413# * logTempK
        lnK1 = lnK1 + (-.20760841# - 4.0484 / TempK) * sqrSal
        lnK1 = lnK1 + .08468345# * Sal - .00654208# * sqrSal * Sal
        K1 = EXP(lnK1): ' this is on the total pH scale in mol/kg-H2O
        K1 = K1 * (1! - .001005 * Sal): ' convert to mol/kg-SW
        K1 = K1 / SWStoTOT: ' convert to SWS pH scale
'
'       This is eq. 30 on p. 254 and what they use in their abstract:
        lnK2 = -9.226508 - 3351.6106# / TempK - .2005743 * logTempK
        lnK2 = lnK2 + (-.106901773# - 23.9722 / TempK) * sqrSal
        lnK2 = lnK2 + .1130822 * Sal - .00846934# * sqrSal * Sal
        K2 = EXP(lnK2): ' this is on the total pH scale in mol/kg-H2O
        K2 = K2 * (1! - .001005 * Sal): ' convert to mol/kg-SW
        K2 = K2 / SWStoTOT: ' convert to SWS pH scale
'
'
'************************************************
        CASE 2: ' GOYET AND POISSON, Deep-Sea Research, 36(11):1635-1654, 1989
'               The 2s precision in pK1 is .011, or 2.5% in K1.
'               The 2s precision in pK2 is .02, or 4.5% in K2.
'
'       This is in Table 5 on p. 1652 and what they use in the abstract:
        pK1 = 812.27 / TempK + 3.356 - .00171 * Sal * logTempK
        pK1 = pK1 + .000091 * Sal * Sal
        K1 = 10! ^ (-pK1): ' this is on the SWS pH scale in mol/kg-SW
'
'       This is in Table 5 on p. 1652 and what they use in the abstract:
        pK2 = 1450.87 / TempK + 4.604 - .00385 * Sal * logTempK
        pK2 = pK2 + .000182 * Sal * Sal
        K2 = 10! ^ (-pK2): ' this is on the SWS pH scale in mol/kg-SW
```

```
'
'
'************************************************
        CASE 3: ' HANSSON refit BY DICKSON AND MILLERO
'               Dickson and Millero, Deep-Sea Research, 34(10):1733-1743, 1987
'               (see also Corrigenda, Deep-Sea Research, 36:983, 1989)
'               refit data of Hansson, Deep-Sea Research, 20:461-478, 1973
'               and Hansson, Acta Chemica Scandanavia, 27:931-944, 1973.
'               on the SWS pH scale in mol/kg-SW.
'               Hansson gave his results on the Total scale (he called it
'                       the seawater scale) and in mol/kg-SW.
'               !!! Typo in DM on p. 1739 in Table 4: the equation for pK2*
'                       for Hansson should have a .000132 *S^2
'                       instead of a .000116 *S^2.
'               The 2s precision in pK1 is .013, or 3% in K1.
'               The 2s precision in pK2 is .017, or 4.1% in K2.
'
'       This is from Table 4 on p. 1739.
        pK1 = 851.4 / TempK + 3.237 - .0106 * Sal + .000105 * Sal * Sal
        K1 = 10! ^ (-pK1): ' this is on the SWS pH scale in mol/kg-SW
'
'       This is from Table 4 on p. 1739.
        pK2 = -3885.4 / TempK + 125.844 - 18.141 * logTempK
        pK2 = pK2 - .0192 * Sal + .000132 * Sal * Sal
        K2 = 10! ^ (-pK2): ' this is on the SWS pH scale in mol/kg-SW
'
'
'************************************************
        CASE 4: ' MEHRBACH refit BY DICKSON AND MILLERO
'               Dickson and Millero, Deep-Sea Research, 34(10):1733-1743, 1987
'               (see also Corrigenda, Deep-Sea Research, 36:983, 1989)
'               refit data of Mehrbach et al, Limn Oc, 18(6):897-907, 1973
'               on the SWS pH scale in mol/kg-SW.
'               Mehrbach et al gave results on the NBS scale.
'               The 2s precision in pK1 is .011, or 2.6% in K1.
'               The 2s precision in pK2 is .020, or 4.6% in K2.
'
'       This is in Table 4 on p. 1739.
        pK1 = 3670.7 / TempK - 62.008 + 9.7944 * logTempK
        pK1 = pK1 - .0118 * Sal + .000116 * Sal * Sal
        K1 = 10! ^ (-pK1): ' this is on the SWS pH scale in mol/kg-SW
'
'       This is in Table 4 on p. 1739.
        pK2 = 1394.7 / TempK + 4.777 - .0184 * Sal + .000118 * Sal * Sal
        K2 = 10! ^ (-pK2): ' this is on the SWS pH scale in mol/kg-SW
'
'
'************************************************
        CASE 5: ' HANSSON and MEHRBACH refit BY DICKSON AND MILLERO
'               Dickson and Millero, Deep-Sea Research,34(10):1733-1743, 1987
'               (see also Corrigenda, Deep-Sea Research, 36:983, 1989)
'               refit data of Hansson, Deep-Sea Research, 20:461-478, 1973,
'               Hansson, Acta Chemica Scandanavia, 27:931-944, 1973,
'               and Mehrbach et al, Limnol. Oceanogr.,18(6):897-907, 1973
'               on the SWS pH scale in mol/kg-SW.
'               !!! Typo in DM on p. 1740 in Table 5: the second equation
'                       should be pK2* =, not pK1* =.
'               The 2s precision in pK1 is .017, or 4% in K1.
'               The 2s precision in pK2 is .026, or 6% in K2.
'
'       This is in Table 5 on p. 1740.
        pK1 = 845! / TempK + 3.248 - .0098 * Sal + .000087 * Sal * Sal
        K1 = 10! ^ (-pK1): ' this is on the SWS pH scale in mol/kg-SW
'
'       This is in Table 5 on p. 1740.
        pK2 = 1377.3 / TempK + 4.824 - .0185 * Sal + .000122 * Sal * Sal
        K2 = 10! ^ (-pK2): ' this is on the SWS pH scale in mol/kg-SW
'
'
'************************************************
        CASE 6, 7:
'               GEOSECS and Peng et al use K1, K2 from Mehrbach et al,
'               Limnology and Oceanography, 18(6):897-907, 1973.
'               The 2s precision in pK1 is .005, or 1.2% in K1.
'               The 2s precision in pK2 is .008, or 2% in K2.
        logK1 = 13.7201 - .031334 * TempK - 3235.76 / TempK
        logK1 = logK1 - .000013 * Sal * TempK + .1032 * sqrSal
        K1 = 10! ^ (logK1): ' this is on the NBS scale
        K1 = K1 / fH: ' convert to SWS scale
'
        logK2 = -5371.9645# - 1.671221 * TempK + 128375.28# / TempK
        logK2 = logK2 + 2194.3055# * logTempK / LOG(10!) - .22913 * Sal
        logK2 = logK2 - 18.3802 * LOG(Sal) / LOG(10!) + .00080944# * Sal * TempK
        logK2 = logK2 + 5617.11 * LOG(Sal) / LOG(10!) / TempK - 2.136 * Sal /
TempK
        K2 = 10! ^ (logK2): ' this is on the NBS scale
        K2 = K2 / fH: ' convert to SWS scale
'
'
'************************************************
        CASE 8
'               Millero, F. J., Geochemica et Cosmochemica Acta 43:1651-1661,
1979:
'               K1 from refit data from Harned and Davis,
'                       J American Chemical Society, 65:2030-2037, 1943.
'               K2 from refit data from Harned and Scholes,
'                       J American Chemical Society, 43:1706-1709, 1941.
'       These are the thermodynamic constants:
        lnK1 = 290.9097 - 14554.21 / TempK - 45.0575 * logTempK
        K1 = EXP(lnK1)
        lnK2 = 207.6548 - 11843.79 / TempK - 33.6485 * logTempK
        K2 = EXP(lnK2)
'
'
'************************************************
        END SELECT
'
'
'***************************************************************************
'***************************************************************************
CorrectKsForPressureNow:
' Currently: For WhichKs% = 1 to 7, all Ks (except KF and KS, which are on
'       the free scale) are on the SWS scale.
'       For WhichKs% = 6, KW set to 0, KP1, KP2, KP3, KSi don't matter.
'       For WhichKs% = 8, K1, K2, and KW are on the "pH" pH scale
'       (the pH scales are the same in this case); the other Ks don't matter.
'
'
' No salinity dependence is given for the pressure coefficients here.
' It is assumed that the salinity is at or very near Sal = 35.
' These are valid for the SWS pH scale, but the difference between this and
' the total only yields a difference of .004 pH units at 1000 bars, much
' less than the uncertainties in the values.
'
'
'****************************************************************************
' The sources used are:
' Millero, 1995:
'       Millero, F. J., Thermodynamics of the carbon dioxide system in the
'       oceans, Geochemica et Cosmochemica Acta 59:661-677, 1995.
'       See table 9 and eqs. 90-92, p. 675.
'       TYPO!!!: a factor of 10^3 was left out of the definition of Kappa
'       TYPO!!!: the value of R given is incorrect with the wrong units
'       TYPO!!!: the values of the a's for H2S and H2O are from the 1983
'                values for fresh water
'       TYPO!!!: the value of a1 for B(OH)3 should be +.1622
'       !!! Table 9 on p. 675 has no values for Si.
'       There are a variety of other typos in Table 9 on p. 675.
'       There are other typos in the paper, and most of the check values
'       given don't check.
' Millero, 1992:
'       Millero, Frank J., and Sohn, Mary L., Chemical Oceanography,
'       CRC Press, 1992. See chapter 6.
'       TYPO!!!: this chapter has numerous typos (eqs. 36, 52, 56, 65, 72,
'               79, and 96 have typos).
' Millero, 1983:
'       Millero, Frank J., Influence of pressure on chemical processes in
'       the sea. Chapter 43 in Chemical Oceanography, eds. Riley, J. P. and
'       Chester, R., Academic Press, 1983.
'       TYPO!!!: p. 51, eq. 94: the value -26.69 should be -25.59
'       TYPO!!!: p. 51, eq. 95: the term .1700t should be .0800t
'       these two are necessary to match the values given in Table 43.24
' Millero, 1979:
'       Millero, F. J., The thermodynamics of the carbon dioxide system
'       in seawater, Geochemica et Cosmochemica Acta 43:1651-1661, 1979.
'       See table 5 and eqs. 7, 7a, 7b on pp. 1656-1657.
' Takahashi et al, in GEOSECS Pacific Expedition, v. 3, 1982.
'       TYPO!!!: the pressure dependence of K2 should have a 16.4, not 26.4
'       This matches the GEOSECS results and is in Edmond and Gieskes.
' Culberson, C. H. and Pytkowicz, R. M., Effect of pressure on carbonic acid,
'       boric acid, and the pH of seawater, Limnology and Oceanography
'       13:403-417, 1968.
' Edmond, John M. and Gieskes, J. M. T. M., The calculation of the degree of
'       seawater with respect to calcium carbonate under in situ conditions,
'       Geochemica et Cosmochemica Acta, 34:1261-1291, 1970.
'
'
'****************************************************************************
' These references often disagree and give different fits for the same thing.
' They are not always just an update either; that is, Millero, 1995 may agree
'       with Millero, 1979, but differ from Millero, 1983.
' For WhichKs% = 7 (Peng choice) I used the same factors for KW, KP1, KP2,
'       KP3, and KSi as for the other cases. Peng et al didn't consider the
'       case of P different from 0. GEOSECS did consider pressure, but didn't
'       include Phos, Si, or OH, so including the factors here won't matter.
' For WhichKs% = 8 (freshwater) the values are from Millero, 1983 (for K1, K2,
'       and KW). The other aren't used (TB = TS = TF = TP = TSi = 0.), so
'       including the factors won't matter.
'
'
'************************************************
CorrectK1K2KBForPressure:
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5
PressureEffectsOnK1:
'               These are from Millero, 1995.
'               They are the same as Millero, 1979 and Millero, 1992.
'               They are from data of Culberson and Pytkowicz, 1968.
                deltaV = -25.5 + .1271 * TempC
                'deltaV = deltaV - .151 * (Sal - 34.8): ' Millero, 1979
                Kappa = (-3.08 + .0877 * TempC) / 1000!
                'Kappa = Kappa  - .578 * (Sal - 34.8)/1000.: ' Millero, 1979
                lnK1fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'               The fits given in Millero, 1983 are somewhat different.
'
'
PressureEffectsOnK2:
'               These are from Millero, 1995.
'               They are the same as Millero, 1979 and Millero, 1992.
'               They are from data of Culberson and Pytkowicz, 1968.
                deltaV = -15.82 - .0219 * TempC
                'deltaV = deltaV + .321 * (Sal - 34.8): ' Millero, 1979
                Kappa = (1.13 - .1475 * TempC) / 1000!
                'Kappa = Kappa - .314 * (Sal - 34.8) / 1000!: ' Millero, 1979
                lnK2fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'               The fit given in Millero, 1983 is different.
'               Not by a lot for deltaV, but by much for Kappa!!!. '
'
'
PressureEffectsOnKB:
'               This is from Millero, 1979.
'               It is from data of Culberson and Pytkowicz, 1968.
                deltaV = -29.48 + .1622 * TempC - .002608 * TempC * TempC
'               Millero, 1983 has:
                'deltaV = -28.56 + .1211 * TempC - .000321 * TempC * TempC
'               Millero, 1992 has:
                'deltaV = -29.48 + .1622 * TempC + .295 * (Sal - 34.8)
'               Millero, 1995 has:
                'deltaV = -29.48 - .1622 * TempC - .002608 * TempC * TempC
                'deltaV = deltaV + .295 * (Sal - 34.8): ' Millero, 1979
                Kappa = -2.84 / 1000!: ' Millero, 1979
'               Millero, 1992 and Millero, 1995 also have this.
                'Kappa = Kappa + .354 * (Sal - 34.8) / 1000!: ' Millero,1979
'               Millero, 1983 has:
                'Kappa = (-3! + .0427 * TempC) / 1000!
                lnKBfac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
```

```
'**************************
        CASE 6, 7
'               GEOSECS Pressure Effects On K1, K2, KB (on the NBS scale)
'               Takahashi et al, GEOSECS Pacific Expedition v. 3, 1982 quotes
'               Culberson and Pytkowicz, L and O 13:403-417, 1968:
'               but the fits are the same as those in
'               Edmond and Gieskes, GCA, 34:1261-1291, 1970
'               who in turn quote Li, personal communication
'
                lnK1fac = (24.2 - .085 * TempC) * Pbar / RT
                lnK2fac = (16.4 - .04 * TempC) * Pbar / RT
'               Takahashi et al had 26.4, but 16.4 is from Edmond and Gieskes
'               and matches the GEOSECS results
                lnKBfac = (27.5 - .095 * TempC) * Pbar / RT
'
'
'**************************
        CASE 8
PressureEffectsOnK1inFreshWater:
'               This is from Millero, 1983.
                deltaV = -30.54 + .1849 * TempC - .0023366 * TempC * TempC
                Kappa = (-6.22 + .1368 * TempC - .001233 * TempC * TempC) /
1000!
                lnK1fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
PressureEffectsOnK2inFreshWater:
'               This is from Millero, 1983.
                deltaV = -29.81 + .115 * TempC - .001816 * TempC * TempC
                Kappa = (-5.74 + .093 * TempC - .001896 * TempC * TempC) / 1000!
                lnK2fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
                lnKBfac = 0!': this doesn't matter since TB = 0 for this case
'
'
        END SELECT
'************************************************
CorrectKWForPressure:
        SELECT CASE WhichKs%
        CASE 1, 2, 3, 4, 5, 6, 7
' GEOSECS doesn't include OH term, so this won't matter.
' Peng et al didn't include pressure, but here I assume that the KW correction
'       is the same as for the other seawater cases.
PressureEffectsOnKW:
'               This is from Millero, 1983 and his programs CO2ROY(T).BAS.
                deltaV = -20.02 + .1119 * TempC - .001409 * TempC * TempC
'               Millero, 1992 and Millero, 1995 have:
                'deltaV = -25.6 + .2324*TempC - .0036246*TempC*TempC
'               This is the freshwater value listed in Millero, 1983.
'               The difference is about 4 to 5 over the range 0 < TempC < 20,
'               which corresponds to a change in KW(P) of 3% at 200 bar,
'               8% at 500 bar, and 18% at 1000 bar.
'               This is probably correct since in Millero, 1983 values of
'               -deltaVs are less in seawater than pure water in all cases.
                Kappa = (-5.13 + .0794 * TempC) / 1000!: ' Millero, 1983
'               Millero, 1995 has this too, but Millero, 1992 is different.
```

```
                lnKWfac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'               Millero, 1979 does not list values for these.
'
'
        CASE 8
PressureEffectsOnKWinFreshWater:
'               This is from Millero, 1983.
                deltaV = -25.6 + .2324 * TempC - .0036246 * TempC * TempC
                Kappa = (-7.33 + .1368 * TempC - .001233 * TempC * TempC) /
1000!
                lnKWfac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'               !!! NOTE the temperature dependence of KappaK1 and KappaKW
'               for fresh water in Millero, 1983 are the same.
'
'
        END SELECT
'***********************************************
PressureEffectsOnKF:
'       This is from Millero, 1995, which is the same as Millero, 1983.
'       It is assumed that KF is on the free pH scale.
        deltaV = -9.78 - .009 * TempC - .000942 * TempC * TempC
        Kappa = (-3.91 + .054 * TempC) / 1000!
        lnKFfac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
PressureEffectsOnKS:
'       This is from Millero, 1995, which is the same as Millero, 1983.
'       It is assumed that KS is on the free pH scale.
        deltaV = -18.03 + .0466 * TempC + .000316 * TempC * TempC
        Kappa = (-4.53 + .09 * TempC) / 1000!
        lnKSfac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
'***********************************************
CorrectKP1KP2KP3KSiForPressure:
' These corrections don't matter for the GEOSECS choice (WhichKs% = 6) and
'       the freshwater choice (WhichKs% = 8). For the Peng choice I assume
'       that they are the same as for the other choices (WhichKs% = 1 to 5).
' The corrections for KP1, KP2, and KP3 are from Millero, 1995, which are the
'       same as Millero, 1983.
'
'
PressureEffectsOnKP1:
        deltaV = -14.51 + .1211 * TempC - .000321 * TempC * TempC
        Kappa = (-2.67 + .0427 * TempC) / 1000!
        lnKP1fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
PressureEffectsOnKP2:
        deltaV = -23.12 + .1758 * TempC - .002647 * TempC * TempC
        Kappa = (-5.15 + .09 * TempC) / 1000!
        lnKP2fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
PressureEffectsOnKP3:
        deltaV = -26.57 + .202 * TempC - .003042 * TempC * TempC
        Kappa = (-4.08 + .0714 * TempC) / 1000!
        lnKP3fac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
PressureEffectsOnKSi:
' !!!! The only mention of this is Millero, 1995 where it is stated that the
'       values have been estimated from the values of boric acid. HOWEVER,
'       there is no listing of the values in the table.
'       I used the values for boric acid from above.
        deltaV = -29.48 + .1622 * TempC - .002608 * TempC * TempC
        Kappa = -2.84 / 1000!
        lnKSifac = (-deltaV + .5 * Kappa * Pbar) * Pbar / RT
'
'
'************************************************
CorrectKsForPressureHere:
        K1fac = EXP(lnK1fac): K1 = K1 * K1fac
        K2fac = EXP(lnK2fac): K2 = K2 * K2fac
        KWfac = EXP(lnKWfac): KW = KW * KWfac
        KBfac = EXP(lnKBfac): KB = KB * KBfac
        KFfac = EXP(lnKFfac): KF = KF * KFfac
        KSfac = EXP(lnKSfac): KS = KS * KSfac
        KP1fac = EXP(lnKP1fac): KP1 = KP1 * KP1fac
        KP2fac = EXP(lnKP2fac): KP2 = KP2 * KP2fac
        KP3fac = EXP(lnKP3fac): KP3 = KP3 * KP3fac
        KSifac = EXP(lnKSifac): KSi = KSi * KSifac
'
'
'****************************************************************************
MakeKMatrix:
        K(1) = K1: K(2) = K2: K(3) = KW: K(4) = KB: K(5) = KF
        K(6) = KS: K(7) = KP1: K(8) = KP2: K(9) = KP3: K(10) = KSi
'
'
'************************************************
CorrectpHScaleConversionsForPressure:
'       fH has been assumed to be independent of pressure.
        SWStoTOT = (1! + TS / KS) / (1! + TS / KS + TF / KF)
        FREEtoTOT = 1! + TS / KS
'       The values KS and KF are already pressure-corrected, so the pH scale
'       conversions are now valid at pressure.
'
'
'************************************************
FindpHScaleConversionFactor:
        SELECT CASE pHScale$: ' this is the scale they will be put on
                CASE "pH"
                        'there are only K1, K2, and KW and they should be ok
                        pHfactor = 1!
                CASE "pHsws"
                        ' they are all on this now
                        pHfactor = 1!
                CASE "pHtot"
                        pHfactor = SWStoTOT
                CASE "pHfree"
                        pHfactor = SWStoTOT / FREEtoTOT
                CASE "pHNBS"
                        pHfactor = fH
        END SELECT
'
'
'*************************************************
ConvertFromSWSpHScaleToChosenScale:
        FOR II% = 1 TO 4
                K(II%) = K(II%) * pHfactor
        NEXT II%
        ' KS and KF remain on the free pH scale
        FOR II% = 7 TO 10
                K(II%) = K(II%) * pHfactor
        NEXT II%
'
'
'*************************************************
' The constants should all be on the chosen pH scale at pressure.
'
'
'****************************************************************************
CalculateFugacityConstants:
' !!! This assumes that the pressure is at one atmosphere, or close to it.
' Otherwise, the Pres term in the exponent affects the results.
'       Weiss, R. F., Marine Chemistry 2:203-215, 1974.
'       Delta and B in cm3/mol
        Delta = (57.7 - .118 * TempK)
        B = -1636.75 + 12.0408 * TempK - .0327957 * TempK * TempK
        B = B + 3.16528 * .00001 * TempK * TempK * TempK
'
'
'       For a mixture of CO2 and air at 1 atm (at low CO2 concentrations):
        P1atm = 1.01325: ' in bar
        FugFac = EXP((B + 2! * Delta) * P1atm / RT)
'
'
'*************************************************
        IF WhichKs% = 6 OR WhichKs% = 7 THEN FugFac = 1!
'       GEOSECS and Peng assume pCO2 = fCO2, or FugFac = 1
'
'
'*****************************************************************************
CalculateVPFac:
' Weiss, R. F., and Price, B. A., Nitrous oxide solubility in water and
'       seawater, Marine Chemistry 8:347-359, 1980.
' They fit the data of Goff and Gratch (1946) with the vapor pressure
'       lowering by sea salt as given by Robinson (1954).
' This fits the more complicated Goff and Gratch, and Robinson equations
'       from 273 to 313 deg K and 0 to 40 Sal with a standard error
'       of .015%, about 5 uatm over this range.
' This may be on IPTS-29 since they didn't mention the temperature scale,
'       and the data of Goff and Gratch came before IPTS-48.
' The references are:
' Goff, J. A. and Gratch, S., Low pressure properties of water from -160 deg
'       to 212 deg F, Transactions of the American Society of Heating and
'       Ventilating Engineers 52:95-122, 1946.
' Robinson, Journal of the Marine Biological Association of the U. K.
'       33:449-455, 1954.
'
```

```
'
'       This is eq. 10 on p. 350.
'       This is in atmospheres.
        VPWP = EXP(24.4543 - 67.4509 * (100! / TempK) - 4.8489 * LOG(TempK /
100!))
        VPCorrWP = EXP(-.000544 * Sal)
        VPSWWP = VPWP * VPCorrWP
        VPFac = 1! - VPSWWP: ' this assumes 1 atmosphere
END SUB
SUB ErrorSub (IError%, Error$)
' SUB ErrorSub, version 03.01, 08-27-97, written by Ernie Lewis.
' Inputs: IError%
' Outputs: Error$
' This provides comments for each QBASIC error.
' When a run-time error occurs in QBASIC, the variable ERR is assigned
'       a value corresponding to the error.
' There are 63 errors, numbered from 1 to 88 except for the following:
'       15, 20, 21, 22, 23, 28, 31, 32, 34, 36, 39, 41, 42, 43, 44, 45, 46,
'       47, 48, 49, 60, 65, 66, 77, 78, 79.
'
'
        SELECT CASE IError%
                CASE 1: Error$ = "NEXT without FOR "
                CASE 2: Error$ = "Syntax error "
                CASE 3: Error$ = "RETURN without GOSUB "
                CASE 4: Error$ = "Out of DATA "
                CASE 5: Error$ = "Illegal function call "
                CASE 6: Error$ = "Overflow "
                CASE 7: Error$ = "Out of memory "
                CASE 8: Error$ = "Label not defined "
                CASE 9: Error$ = "Subscript out of range "
                CASE 10: Error$ = "Duplicate definition "
                CASE 11: Error$ = "Division by zero "
                CASE 12: Error$ = "Illegal in direct mode "
                CASE 13: Error$ = "Type mismatch "
                CASE 14: Error$ = "Out of string space "
                CASE 16: Error$ = "String formula too complex "
                CASE 17: Error$ = "Cannot continue "
                CASE 18: Error$ = "Function not defined "
                CASE 19: Error$ = "No RESUME "
                CASE 20: Error$ = "RESUME without error "
                CASE 24: Error$ = "Device timeout "
                CASE 25: Error$ = "Device fault "
                CASE 26: Error$ = "FOR without NEXT "
                CASE 27: Error$ = "Out of paper "
                CASE 29: Error$ = "WHILE without WEND "
                CASE 30: Error$ = "WEND without WHILE "
                CASE 33: Error$ = "Duplicate LABEL "
                CASE 35: Error$ = "Subprogram not defined "
                CASE 37: Error$ = "Argument-count mismatch "
                CASE 38: Error$ = "Array not defined "
                CASE 40: Error$ = "Variable required "
                CASE 50: Error$ = "FIELD overflow "
                CASE 51: Error$ = "Internal error "
                CASE 52: Error$ = "Bad filename or number "
                CASE 53: Error$ = "File not found "
                CASE 54: Error$ = "Bad file mode "
                CASE 55: Error$ = "File already open "
                CASE 56: Error$ = "FIELD statement active "
                CASE 57: Error$ = "Device I/O error "
                CASE 58: Error$ = "File already exists "
                CASE 59: Error$ = "Bad record length "
                CASE 61: Error$ = "Disk full "
                CASE 62: Error$ = "Input past end of file "
                CASE 63: Error$ = "Bad record number "
                CASE 64: Error$ = "Bad filename "
                CASE 67: Error$ = "Too many files "
                CASE 68: Error$ = "Device unavailable "
                CASE 69: Error$ = "Communication-buffer overflow "
                CASE 70: Error$ = "Permission denied "
                CASE 71: Error$ = "Disk not ready "
                CASE 72: Error$ = "Disk-media error "
                CASE 73: Error$ = "Feature unavailable "
                CASE 74: Error$ = "Rename across disks "
                CASE 75: Error$ = "Path/File access error "
                CASE 76: Error$ = "Path not found "
                CASE 80: Error$ = "Feature removed "
                CASE 81: Error$ = "Invalid name "
                CASE 82: Error$ = "Table not found "
                CASE 83: Error$ = "Index not found "
                CASE 84: Error$ = "Invalid column "
                CASE 85: Error$ = "No current record "
                CASE 86: Error$ = "Duplicate value for unique index "
                CASE 87: Error$ = "Invalid operation on null index "
                CASE 88: Error$ = "Database needs repair "
                CASE ELSE: Error$ = ""
        END SELECT
END SUB
SUB FindpHfCO2fromTATC (pHScale$, WhichKs%, WhoseKSO4%, TA, TC, Sal, K(), T(),
TempC, Pdbar, pH, fCO2, pCO2)
' SUB FindpHfCO2fromTATC, version 01.02, 10-10-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, TA, TC, Sal, K(), T(), TempC, Pdbar
' Outputs: pH, fCO2, pCO2
' This calculates pH, fCO2, and pCO2 from TA and TC at output conditions.
'
'
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
'
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatepHfromTATC(TA, TC, K(), T(), pH)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2): pCO2 = fCO2 /
FugFac
END SUB
SUB FindpHOnAllScales (pHScale$, pH, K(), T(), fH, pHNBS, pHfree, pHtot, pHsws)
' SUB FindpHOnAllScales, version 01.02, 01-08-97, written by Ernie Lewis.
' Inputs: pHScale$, pH, K(), T(), fH
' Outputs: pHNBS, pHfree, pHTot, pHSWS
' This takes the pH on the given scale and finds the pH on all scales.
'
'
        TS = T(3): TF = T(2)
        KS = K(6): KF = K(5): 'these are at the given T, S, P
        FREEtoTOT = (1! + TS / KS): ' pH scale conversion factor
        SWStoTOT = (1! + TS / KS) / (1! + TS / KS + TF / KF): ' pH scale
conversion factor
        SELECT CASE pHScale$: ' this is the pH scale pH is on now
                CASE "pHNBS"
                        factor = -LOG(SWStoTOT) / LOG(.1) + LOG(fH) / LOG(.1)
                CASE "pHfree"
                        factor = -LOG(FREEtoTOT) / LOG(.1)
                CASE "pHtot"
                        factor = 0!
                CASE "pHsws"
                        factor = -LOG(SWStoTOT) / LOG(.1)
        END SELECT
        pHtot = pH - factor: ' pH comes into this sub on the given scale
        pHNBS = pHtot - LOG(SWStoTOT) / LOG(.1) + LOG(fH) / LOG(.1)
        pHfree = pHtot - LOG(FREEtoTOT) / LOG(.1)
        pHsws = pHtot - LOG(SWStoTOT) / LOG(.1)
END SUB
SUB InputfCO2 (fCO2)
' SUB InputfCO2, version 01.01, 10-10-97, written by Ernie Lewis.
' Inputs: fCO2
' Outputs: fCO2
' This allows the user to input the value of fCO2.
'
'
        RR = CSRLIN: CC = POS(0)
        PRINT USING "    Enter fCO2 in uatm (####.#): "; fCO2 * 1000000!;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", fCO2$: IF fCO2$ <> "" THEN fCO2 = VAL(fCO2$) / 1000000!
        LOCATE RR, 1: PRINT USING "    fCO2 = ####.# "; fCO2 * 1000000!;
        PRINT SPACE$(40)
END SUB
SUB InputFileInfo (NHeaderLines%, NIDFields%, MVD, InputFile$, OutputFile$,
MVFlag$)
' SUB InputFileInfo, version 02.02, 03-11-97, written by Ernie Lewis.
' Inputs: NHeaderLines%, NIDFields%, MVD, MVFlag$
' Outputs: NHeaderLines%, NIDFields%, MVD, InputFile$, OutputFile$, MVFlag$
' This takes the user input names for the input and output files, the number
'       of header lines on the input files, the number of ID fields for each
'       entry, a value to denote missing data, and whether or not to have a
'       flag to denote which samples had missing data.
'
'
TopOfInputFileInfo:
        CLS
'
'
EnterInputFileName:
        LOCATE 5, 1
        PRINT "    Enter input file name:                "
        PRINT "    (<enter> will exit the program, "
        PRINT "    .INP assumed if no extension given). "
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 5, 28: INPUT "", InputFile$
        IF InputFile$ = "" THEN CLS : END
'
```

```
'       This makes sure there are no blanks in InputFile$.
        IF INSTR(InputFile$, " ") <> 0 THEN
                BEEP
                PRINT "    No spaces allowed !!!!!!!!!       "
                PRINT SPACE$(80)
                SLEEP 3
                GOTO EnterInputFileName:
        END IF
'
        where% = INSTR(InputFile$, ".")
        IF where% = 0 THEN
                InputFileRoot$ = InputFile$
                InputFile$ = InputFileRoot$ + ".INP"
        ELSE
                InputFileRoot$ = LEFT$(InputFile$, where% - 1)
        END IF
        InputFile$ = UCASE$(InputFile$)
        LOCATE 5, 1: PRINT SPACE$(80): PRINT SPACE$(80): PRINT SPACE$(80)
        LOCATE 5, 1: PRINT "    input filename:  "; InputFile$
'
'
EnterOutputFileName:
        LOCATE 7, 1
        PRINT "    Enter output file name: "
        PRINT "    (<enter> assumes same root as input file, "
        PRINT "    .OUT assumed if no extension given). "
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 7, 29: INPUT "", OutputFile$
        IF OutputFile$ = "" THEN OutputFile$ = InputFileRoot$ + ".OUT"
        IF INSTR(OutputFile$, ".") = 0 THEN
                SELECT CASE OutputFile$
                        CASE "CON", "SCRN", "PRN", "COM1", "COM2", "LPT1",
"LPT2", "LPT3"
                        CASE ELSE
                                OutputFile$ = OutputFile$ + ".OUT"
                END SELECT
        END IF
        OutputFile$ = UCASE$(OutputFile$)
        LOCATE 7, 1: PRINT SPACE$(80): PRINT SPACE$(80): PRINT SPACE$(80)
        LOCATE 7, 1: PRINT "    output filename: ";
        IF OutputFile$ = "CON" THEN
                PRINT "the output will be printed to the screen. "
        ELSE
                PRINT OutputFile$
        END IF
'
'
EnterNumberOfHeaderLines:
        LOCATE 9, 1
        PRINT "    Enter the number of header lines before the data start (";
        PRINT LTRIM$(RTRIM$(STR$(NHeaderLines%)));
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "): ", NHeaderLines$
        IF NHeaderLines$ <> "" THEN NHeaderLines% = VAL(NHeaderLines$)
        LOCATE 9, 1: PRINT SPACE$(80): LOCATE 9, 1
        PRINT "    There will be"; NHeaderLines%; "header lines in the input
file. "
```

```
'
'
EnterNumberOfIDFields:
        LOCATE 11, 1
        PRINT "    Enter the number of ID fields for each sample (";
        PRINT LTRIM$(RTRIM$(STR$(NIDFields%)));
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "): ", NIDFields$
        IF NIDFields$ <> "" THEN NIDFields% = VAL(NIDFields$)
        IF NIDFields% < 1 THEN NIDFields% = 0
        LOCATE 11, 1: PRINT SPACE$(80): LOCATE 11, 1
        PRINT "    There will be"; NIDFields%; "ID fields for each entry. "
'
'
EnterMVD:
        LOCATE 14, 1
        PRINT "    (this value should never otherwise occur in any inputs). "
        LOCATE 13, 1: PRINT "    Enter a value to denote missing data (";
        PRINT RTRIM$(STR$(MVD));
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "): ", MVD$
        IF MVD$ <> "" THEN MVD = VAL(MVD$)
        LOCATE 13, 1: PRINT SPACE$(80): PRINT SPACE$(80)
        LOCATE 13, 1: PRINT "    The missing value designator is "; MVD
'
'
EnterChoiceOfMissingValueFlag:
        LOCATE 16, 1
        PRINT "         This will be another field in the output file containing
"; MVD
        PRINT "         if there are missing data values for a sample, 0
otherwise. "
        LOCATE 15, 1: PRINT USING "    Do you want a missing value flag? (&): ";
MVFlag$;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", Q$
        LOCATE 15, 1: PRINT SPACE$(80): PRINT SPACE$(80): PRINT SPACE$(80)
        LOCATE 15, 1
        IF Q$ = "" THEN Q$ = MVFlag$
        SELECT CASE Q$
                CASE "Y", "y"
                        MVFlag$ = "Y"
                        PRINT "    There will be a missing value flag of "; MVD
                CASE "N", "n"
                        MVFlag$ = "N"
                        PRINT "    There will not be a missing value flag. "
                CASE ELSE
                        GOTO EnterChoiceOfMissingValueFlag:
        END SELECT
'
'
ReviewEntriesForBatchFileInfo:
        LOCATE 20, 1
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Are these the desired choices? (Y or N): ", Q$
        IF Q$ = "Y" OR Q$ = "y" THEN EXIT SUB
        IF Q$ = "N" OR Q$ = "n" THEN GOTO TopOfInputFileInfo:
```

```
        GOTO ReviewEntriesForBatchFileInfo:
END SUB
SUB InputFromBatchFile (NIDFields%, MVD, MVFlag$, BatchFlag%, Sal, TP, TSi,
TempCinp, Pdbarinp, TempCout, Pdbarout, Param1, Param2, MVFlag)
' SUB InputFromBatchFile, version 02.01, 03-11-97, written by Ernie Lewis.
' Inputs: NIDFields%, MVD, MVFlag$, BatchFlag%
' Outputs: Sal, TP, TSi, TempCinp, Pdbarinp, TempCout, Pdbarout
' Outputs: Param1, Param2, MVFlag
' This reads a line of data from the batch file.
' If there is missing data then default values are used.
' If BatchFlag% is set to 1, then the program will enter another value.
'       I don't do it here in case there is an EOF just after a missing value.
'
'
        AAA$ = ""
        C$ = ","
'
'
ReadIDFieldsFromBatchFile:
        IF NIDFields% > 0 THEN
                FOR I% = 1 TO NIDFields%
                        INPUT #3, ID$
                        AAA$ = AAA$ + ID$ + C$
                NEXT I%
        END IF
'
'
ReadOtherInformationFromBatchFile:
        INPUT #3, Sal, TP, TSi, TempCinp, Pdbarinp, TempCout, Pdbarout, Param1,
Param2
        AAA$ = AAA$ + STR$(Sal) + C$ + STR$(TP) + C$ + STR$(TSi) + C$
        AAA$ = AAA$ + STR$(TempCinp) + C$ + STR$(Pdbarinp) + C$
        AAA$ = AAA$ + STR$(TempCout) + C$ + STR$(Pdbarout) + C$
        AAA$ = AAA$ + STR$(Param1) + C$ + STR$(Param2) + C$
        PRINT #4, AAA$;
        PRINT AAA$;
        ' PRINT doesn't include "" on strings, WRITE does, so I want PRINT
'
'
CheckForMissingValues:
        'recall MVD is the missing value designator
        MVFlag = 0: ' this says there are no missing values
        IF Sal = MVD THEN Sal = 35!: MVFlag = MVD
        IF TP = MVD THEN TP = 0!: MVFlag = MVD
        TP = TP / 1000000!
        IF TSi = MVD THEN TSi = 0!: MVFlag = MVD
        TSi = TSi / 1000000!
        IF TempCinp = MVD THEN TempC1 = 20!: MVFlag = MVD
        IF Pdbarinp = MVD THEN Pdbarinp = 0!: MVFlag = MVD
        IF TempCout = MVD THEN TempCout = TempCinp: MVFlag = MVD
        IF Pdbarout = MVD THEN Pdbarout = Pdbarinp: MVFlag = MVD
        IF Param1 = MVD OR Param2 = MVD THEN
                MVFlag = MVD
                IF MVFlag$ = "Y" THEN
                        WRITE #4, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD,
MVD, MVD, MVD, MVD
```

```
                        WRITE MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD,
MVD, MVD, MVD
                ELSE
                        WRITE #4, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD,
MVD, MVD, MVD
                        WRITE MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD, MVD,
MVD, MVD
                END IF
                BatchFlag% = 1
        END IF
END SUB
SUB InputParameters (WhichKs%, Sal, TP, TSi, TempCinp, Pdbarinp, TempCout,
Pdbarout)
' SUB InputParameters, version 01.03, 03-11-97, written by Ernie Lewis.
' Inputs: WhichKs%, Sal, TP, TSi, TempCinp, Pdbarinp, TempCout, Pdbarout
' Outputs: WhichKs%, Sal, TP, TSi, TempCinp, Pdbarinp, TempCout, Pdbarout
' This allows the user to input the parameters to be used in the program.
'
'
TopOfInputParameters:
        CLS : PRINT : PRINT
        IF WhichKs% <> 8 THEN
                PRINT USING "    Enter Sal (##.##): "; Sal
                IF WhichKs% <> 6 THEN
                        PRINT USING "    Enter total phosphate in umol/kg-SW
(##.#): "; TP * 1000000!
                        PRINT USING "    Enter total silicate in umol/kg-SW
(###.#): "; TSi * 1000000!
                ELSE
                        PRINT "    GEOSECS did not include the effects of
phosphate and silicate. "
                        TP = 0!: TSi = 0!
                        PRINT
                END IF
        ELSE
                Sal = 0!: TP = 0!: TSi = 0!
                PRINT : PRINT : PRINT
        END IF
        PRINT
        PRINT USING "    Enter input temperature in deg C (##.##): "; TempCinp
        PRINT USING "    Enter input pressure in dbar or depth in meters
(#####): "; Pdbarinp
        PRINT
        PRINT USING "    Enter output temperature in deg C (##.##): "; TempCout
        PRINT USING "    Enter output pressure in dbar or depth in meters
(#####): "; Pdbarout
'
'
        IF WhichKs% = 8 THEN Sal = 0!: TP = 0!: TSi = 0!: GOTO InputTempCinp:
InputSal:
        LOCATE 3, 1: PRINT USING "    Enter Sal (##.##): "; Sal;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", Sal$
        IF Sal$ <> "" THEN Saltry = VAL(Sal$) ELSE Saltry = Sal
        IF Saltry < 20! OR Saltry > 40! THEN
                LOCATE 15, 1
                PRINT "    Most of the fits of the constants are only valid for
20 < Sal < 40. "
                PRINT "    The validity of the results are compromised by the
use of this Sal. "
                DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
                INPUT "    Enter <K> to keep, anything else to re-enter: ", Q$
                LOCATE 15, 1
                PRINT SPACE$(80): PRINT SPACE$(80): PRINT SPACE$(80)
                IF Q$ <> "K" AND Q$ <> "k" THEN
                        GOTO TopOfInputParameters:
                END IF
        END IF
        Sal = Saltry
        LOCATE 3, 1: PRINT USING "    Sal  =  ##.## "; Sal; : PRINT SPACE$(40)
'
'
        IF WhichKs% = 6 THEN
                TP = 0!: TSi = 0!
                GOTO InputTempCinp:
        END IF
InputPhosphate:
        LOCATE 4, 1: PRINT USING "    Enter total phosphate in umol/kg-SW
(##.#): "; TP * 1000000!;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", TP$
        IF TP$ <> "" THEN TP = VAL(TP$) / 1000000!
        LOCATE 4, 1: PRINT USING "    Phos =  ###.# "; TP * 1000000!; : PRINT
SPACE$(40)
'
'
InputSilicate:
        LOCATE 5, 1: PRINT USING "    Enter total silicate in umol/kg-SW
(###.#): "; TSi * 1000000!;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", TSi$
        IF TSi$ <> "" THEN TSi = VAL(TSi$) / 1000000!
        LOCATE 5, 1: PRINT USING "    Sili =  ###.# "; TSi * 1000000!; : PRINT
SPACE$(40)
'
'
InputTempCinp:
        LOCATE 7, 1: PRINT USING "    Enter input temperature in deg C (##.##):
"; TempCinp;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", TempCinp$
        IF TempCinp$ <> "" THEN TempCinp = VAL(TempCinp$)
        LOCATE 7, 1: PRINT USING "    Input Temp  = ##.## C"; TempCinp; : PRINT
SPACE$(40)
'
'
InputPdbarinp:
        LOCATE 8, 1: PRINT USING "    Enter input pressure in dbar or depth in
meters (#####): "; Pdbarinp;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", Pdbarinp$
        IF Pdbarinp$ <> "" THEN Pdbarinp = VAL(Pdbarinp$)
        LOCATE 8, 1: PRINT USING "    Input Pres  = ##### dbar "; Pdbarinp; :
PRINT SPACE$(45)
'
'
InputTempCout:
        LOCATE 10, 1: PRINT USING "    Enter output temperature in deg C
(##.##): "; TempCout;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", TempCout$
        IF TempCout$ <> "" THEN TempCout = VAL(TempCout$)
        LOCATE 10, 1: PRINT USING "    Output Temp = ##.## C "; TempCout; :
PRINT SPACE$(40)
'
'
InputPdbarout:
        LOCATE 11, 1: PRINT USING "    Enter output pressure in dbar or depth in
meters (#####): "; Pdbarout;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", Pdbarout$
        IF Pdbarout$ <> "" THEN Pdbarout = VAL(Pdbarout$)
        LOCATE 11, 1: PRINT USING "    Output Pres = ##### dbar "; Pdbarout; :
PRINT SPACE$(45)
'
'
        PRINT : PRINT : PRINT
END SUB
SUB InputpCO2 (pCO2)
' SUB InputpCO2, version 01.01, 10-10-97, written by Ernie Lewis.
' Inputs: pCO2
' Outputs: pCO2
' This allows the user to input the value of pCO2.
'
'
        RR = CSRLIN: CC = POS(0)
        PRINT USING "    Enter pCO2 in uatm (####.#): "; pCO2 * 1000000!;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", pCO2$: IF pCO2$ <> "" THEN pCO2 = VAL(pCO2$) / 1000000!
        LOCATE RR, 1: PRINT USING "    pCO2 = ####.# "; pCO2 * 1000000!;
        PRINT SPACE$(40)
END SUB
SUB InputpH (pHScale$, pH)
' SUB InputpH, version 01.02, 05-23-97, written by Ernie Lewis.
' Inputs: pHScale$, pH
' Outputs: pH
' This allows the user to input the value of pH.
'
'
        RR = CSRLIN: CC = POS(0)
        SELECT CASE pHScale$
                CASE "pHtot"
                        PRINT USING "    Enter pH on the total scale (#.####):
"; pH;
                CASE "pHsws"
                        PRINT USING "    Enter pH on the seawater scale
(#.####): "; pH;
                CASE "pHfree"
```

```
                        PRINT USING "    Enter pH on the free scale (#.####): ";
pH;
                CASE "pHNBS"
                        PRINT USING "    Enter pH on the NBS scale (#.####): ";
pH;
                CASE "pH"
                        PRINT USING "    Enter the pH (#.####): "; pH;
                        ' this is the freshwater case and the scales are the
same
        END SELECT
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", pH$: IF pH$ <> "" THEN pH = VAL(pH$)
        LOCATE RR, 1: PRINT USING "    pH   = #.####      "; pH;
        PRINT SPACE$(40)
END SUB
SUB InputTA (TA)
' SUB InputTA, version 01.00, 11-18-96, written by Ernie Lewis.
' Inputs: TA
' Outputs: TA
' This allows the user to input the value of TA.
'
'
        RR = CSRLIN: CC = POS(0)
        PRINT USING "    Enter TA in umol/kg-SW (####.#): "; TA * 1000000!;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", TA$: IF TA$ <> "" THEN TA = VAL(TA$) / 1000000!
        LOCATE RR, 1: PRINT USING "    TA   = ####.# "; TA * 1000000!;
        PRINT SPACE$(40)
END SUB
SUB InputTC (TC)
' SUB InputTC, version 01.00, 11-18-96, written by Ernie Lewis.
' Inputs: TC
' Outputs: TC
' This allows the user to input the value of TC.
'
'
        RR = CSRLIN: CC = POS(0)
        PRINT USING "    Enter TC in umol/kg-SW (####.#): "; TC * 1000000!;
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "", TC$: IF TC$ <> "" THEN TC = VAL(TC$) / 1000000!
        LOCATE RR, 1: PRINT USING "    TC   = ####.# "; TC * 1000000!;
        PRINT SPACE$(40)
END SUB
SUB PrintFirstScreen
' SUB PrintFirstScreen, version 01.05, 10-15-97, written by Ernie Lewis.
' Inputs: none
' Outputs: none
' This draws the first screen.
'
'
        CLS
        PRINT "    Program CO2SYS, version 01.05, written by Ernie Lewis. "
        PRINT
        PRINT
        PRINT
        PRINT
        PRINT
        PRINT "       CCCC       OOOOO        222        SSS      YY      YY
SSS   "
        PRINT "      CC   C     OO   OO     22   22    SS   SS     YY    YY
SS   SS "
        PRINT "     CC         OO     OO          22    SS           YY  YY
SS      "
        PRINT "     CC         OO     OO        22        S           YY
SSS   "
        PRINT "     CC         OO     OO       22           SS        YY
SS "
        PRINT "      CC   C     OO   OO      22        SS   SS        YY
SS   SS "
        PRINT "       CCCC       OOOOO      2222222      SSS          YY
SSS   "
        PRINT
        PRINT
        PRINT
        PRINT
        PRINT
        PRINT "    Lasciate ogni speranza, voi ch' entrate! "
        '(All hope abandon, ye who enter here.)
        PRINT "                                     Dante, Inferno iii, 9 "
        PRINT "                                     sign on the entrance gates
of hell "
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LOCATE 24, 25: INPUT "Hit <enter> to continue. ", Q$
END SUB
SUB PrintfpCO2OmegasRevelle (WhichKs%, fCO2inp, pCO2inp, xCO2dryinp, Revelleinp,
OmegaCainp, OmegaArinp, fCO2out, pCO2out, xCO2dryout, Revelleout, OmegaCaout,
OmegaArout)
' SUB PrintfpCO2OmegasRevelle, version 01.03, 10-10-97, written by Ernie Lewis.
' Inputs: WhichKs%
' Inputs: fCO2inp, pCO2inp, xCO2dryinp, Revelleinp, OmegaCainp, OmegaArinp
' Inputs: fCO2out, pCO2out, xCO2dryout, Revelleout, OmegaCaout, OmegaArout
' Outputs: none
' This prints fCO2 and pCO2, xCO2 in dry air assuming pTot = 1 atm,
'       omegas for calcite and aragonite, and the Revelle factor, for both
'       input and output conditions.
'
'
        AA$ = "               ####.#                   ####.# "
        BB$ = "        ##.##                    ##.## "
'
'
        PRINT USING "    fCO2 (uatm) " + AA$; fCO2inp * 1000000!; fCO2out *
1000000!
        PRINT USING "    pCO2 (uatm) " + AA$; pCO2inp * 1000000!; pCO2out *
1000000!
        PRINT USING "    xCO2 in dry air at 1 atm   ####.# ppm
####.# ppm "; xCO2dryinp * 1000000!; xCO2dryout * 1000000!
        PRINT
        IF WhichKs% = 6 THEN PRINT "    GEOSECS does not distinguish between
fCO2 and pCO2. ": PRINT
        IF WhichKs% = 7 THEN PRINT "    Peng et al do not distinguish between
fCO2 and pCO2. ": PRINT
        PRINT
        PRINT USING "    Revelle factor      " + BB$; Revelleinp; Revelleout
        PRINT
        PRINT
        IF WhichKs% <> 8 THEN
                PRINT USING "    Omega for calcite   " + BB$; OmegaCainp;
OmegaCaout
                PRINT USING "    Omega for aragonite " + BB$; OmegaArinp;
OmegaArout
        END IF
END SUB
SUB PrintHeader (ICase%, pHScale$, fORp$, TA, TC, pHinp, fCO2inp, pCO2inp, TP,
TSi, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout)
' SUB PrintHeader, version 01.02, 01-03-97, written by Ernie Lewis.
' Inputs: ICase%, pHScale$, fORp$, TA, TC, pHinp, fCO2inp, pCO2inp, TP, TSi, Sal
' Inputs: TempCinp, Pdbarinp, TempCout, Pdbarout
' Outputs: none
' This prints the top of the output page.
'
'
        SELECT CASE pHScale$
                CASE "pHtot"
                        PP$ = "on the total scale "
                CASE "pHsws"
                        PP$ = "on the seawater scale "
                CASE "pHfree"
                        PP$ = "on the free scale "
                CASE "pHNBS"
                        PP$ = "on the NBS scale "
                CASE "pH"
                        PP$ = ""
        END SELECT
'
'
        CLS
        SELECT CASE ICase%
                CASE 1: ' input TA, TC
                        PRINT USING "TA = ####.# umol/kg-SW "; TA * 1000000!
                        PRINT USING "TC = ####.# umol/kg-SW "; TC * 1000000!
                CASE 2: ' input TA, pH
                        PRINT USING "TA = ####.# umol/kg-SW "; TA * 1000000!
                        PRINT USING "pH = #.#### "; pHinp;
                        PRINT PP$
                CASE 3: ' input TA, fCO2 or pCO2
                        PRINT USING "TA = ####.# umol/kg-SW "; TA * 1000000!
                        IF fORp$ = "f" THEN PRINT USING "fCO2 = ####.# uatm ";
fCO2inp * 1000000!
                        IF fORp$ = "p" THEN PRINT USING "pCO2 = ####.# uatm ";
pCO2inp * 1000000!
                CASE 4: ' input TC, pH
                        PRINT USING "TC = ####.# umol/kg-SW "; TC * 1000000!
                        PRINT USING "pH = #.#### "; pHinp;
                        PRINT PP$
                CASE 5: ' input TC, fCO2 or pCO2
                        PRINT USING "TC = ####.# umol/kg-SW "; TC * 1000000!
                        IF fORp$ = "f" THEN PRINT USING "fCO2 = ####.# uatm ";
fCO2inp * 1000000!
                        IF fORp$ = "p" THEN PRINT USING "pCO2 = ####.# uatm ";
pCO2inp * 1000000!
```

```
                CASE 6: ' input pH, fCO2 or pCO2
                        PRINT USING "pH = #.#### "; pHinp;
                        PRINT PP$
                        IF fORp$ = "f" THEN PRINT USING "fCO2 = ####.# uatm ";
fCO2inp * 1000000!
                        IF fORp$ = "p" THEN PRINT USING "pCO2 = ####.# uatm ";
pCO2inp * 1000000!
        END SELECT
        PRINT USING "Sal = ##.## "; Sal
'
'
        ' this covers all cases except WhichKs% = 8; pHScale$ carries this info
        IF pHScale$ <> "pH" THEN
                LOCATE 1, 53: PRINT USING "Phos =  ##.# umol/kg-SW "; TP *
1000000!
                LOCATE 2, 53: PRINT USING "Sili = ###.# umol/kg-SW "; TSi *
1000000!
                LOCATE 3, 53: PRINT ""
        END IF
        PRINT STRING$(80, "-")
        PRINT SPACE$(27);
        PRINT "Input conditions:        Output conditions: "
        PRINT SPACE$(27);
        PRINT USING "Temp = ##.## deg C       Temp = ##.## deg C "; TempCinp;
TempCout
        PRINT SPACE$(27);
        PRINT USING "Pres = ##### dbar        Pres = ##### dbar "; Pdbarinp;
Pdbarout
        PRINT
END SUB
SUB PrintHeaderOnOutputFile (WhichKs%, WhoseKSO4%, fORp$, pHScale$, ICase%,
InputFile$, OutputFile$, NIDFields%, MVFlag$)
' SUB PrintHeaderOnOutputFile, version 02.02, 03-12-97, written by Ernie Lewis.
' Inputs: WhichKs%, WhoseKSO4%, fORp$, pHScale$, ICase%
' Inputs: InputFile$, OutputFile$, NIDFields%, MVFlag$
' Outputs: none
' This prints a header on the output file.
'
'
        SELECT CASE WhichKs%
                CASE 1
                        WhoseConstants$ = "Roy et al"
                CASE 2
                        WhoseConstants$ = "Goyet and Poisson"
                CASE 3
                        WhoseConstants$ = "Hansson refit by Dickson and Millero"
                CASE 4
                        WhoseConstants$ = "Mehrbach et al refit by Dickson and
Millero"
                CASE 5
                        WhoseConstants$ = "Hansson and Mehrbach et al refit by
Dickson and Millero"
                CASE 6
                        WhoseConstants$ = "GEOSECS"
                CASE 7
                        WhoseConstants$ = "Peng et al"
                CASE 8
```

```
                        WhoseConstants$ = "freshwater"
        END SELECT
'
'
        SELECT CASE WhoseKSO4%
                CASE 1
                        WhoseKSO4$ = "Dickson"
                CASE 2
                        WhoseKSO4$ = "Khoo et al"
        END SELECT
'
'
        SELECT CASE pHScale$
                CASE "pHtot"
                        WhichpHScale$ = "total"
                CASE "pHsws"
                        WhichpHScale$ = "seawater"
                CASE "pHfree"
                        WhichpHScale$ = "free"
                CASE "pHNBS"
                        WhichpHScale$ = "NBS"
        END SELECT
'
'
        CLS
        PRINT "The following lines are printed on file "; OutputFile$; ": "
PrintHeaderOnOutputFile:
        PRINT "CO2SYS output from input file "; InputFile$; " on "; DATE$
        PRINT #4, "CO2SYS output from input file "; InputFile$; " on "; DATE$
        PRINT "Constants: "; WhoseConstants$;
        PRINT #4, "Constants: "; WhoseConstants$;
        IF pHScale$ <> "pH" THEN
                PRINT "; KSO4 from "; WhoseKSO4$;
                PRINT #4, "; KSO4 from "; WhoseKSO4$;
                PRINT "; pH on the "; WhichpHScale$; " scale "
                PRINT #4, "; pH on the "; WhichpHScale$; " scale "
        ELSE
                PRINT ""
                PRINT #4, ""
        END IF
        PRINT
        PRINT "Enter other lines here to be printed on the output file (<enter>
to stop): "
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        LINE INPUT UserInputLine$
        DO WHILE UserInputLine$ <> ""
                PRINT #4, UserInputLine$
                LINE INPUT UserInputLine$
        LOOP
'
'
PrintTitleOnOutputFile:
        IF NIDFields% > 0 THEN
                FOR I% = 1 TO NIDFields%
                        PRINT #4, "ID" + LTRIM$(RTRIM$(STR$(I%))) + ", ";
                        PRINT "ID" + LTRIM$(RTRIM$(STR$(I%))) + ", ";
                NEXT I%
        END IF
        PRINT #4, "Sal, TP, TSi, Tinp, Pinp, Tout, Pout, ";
        PRINT "Sal, TP, TSi, Tinp, Pinp, Tout, Pout, ";
        SELECT CASE fORp$
        CASE "f"
                SELECT CASE ICase%
                CASE 1
                        PRINT #4, "TA, TC, pHinp, fCO2inp, ";
                        PRINT "TA, TC, pHinp, fCO2inp, ";
                CASE 2
                        PRINT #4, "TA, pHinp, TC, fCO2inp, ";
                        PRINT "TA, pHinp, TC, fCO2inp, ";
                CASE 3
                        PRINT #4, "TA, fCO2inp, TC, pHinp, ";
                        PRINT "TA, fCO2inp, TC, pHinp, ";
                CASE 4
                        PRINT #4, "TC, pHinp, TA, fCO2inp, ";
                        PRINT "TC, pHinp, TA, fCO2inp, ";
                CASE 5
                        PRINT #4, "TC, fCO2inp, TA, pHinp, ";
                        PRINT "TC, fCO2inp, TA, pHinp, ";
                CASE 6
                        PRINT #4, "pHinp, fCO2inp, TA, TC, ";
                        PRINT "pHinp, fCO2inp, TA, TC, ";
                END SELECT
                PRINT #4, "pHout, fCO2out, OmegaCainp, OmegaArinp, OmegaCaout,
OmegaArout, HCO3inp,CO3inp, HCO3out, CO3out";
                PRINT "pHout, fCO2out, OmegaCainp, OmegaArinp, OmegaCaout,
OmegaArout, HCO3inp,CO3inp, HCO3out, CO3out";
        CASE "p"
                SELECT CASE ICase%
                CASE 1
                        PRINT #4, "TA, TC, pHinp, pCO2inp, ";
                        PRINT "TA, TC, pHinp, pCO2inp, ";
                CASE 2
                        PRINT #4, "TA, pHinp, TC, pCO2inp, ";
                        PRINT "TA, pHinp, TC, pCO2inp, ";
                CASE 3
                        PRINT #4, "TA, pCO2inp, TC, pHinp, ";
                        PRINT "TA, pCO2inp, TC, pHinp, ";
                CASE 4
                        PRINT #4, "TC, pHinp, TA, pCO2inp, ";
                        PRINT "TC, pHinp, TA, pCO2inp, ";
                CASE 5
                        PRINT #4, "TC, pCO2inp, TA, pHinp, ";
                        PRINT "TC, pCO2inp, TA, pHinp, ";
                CASE 6
                        PRINT #4, "pHinp, pCO2inp, TA, TC, ";
                        PRINT "pHinp, pCO2inp, TA, TC, ";
                END SELECT
                PRINT #4, "pHout, pCO2out, OmegaCainp, OmegaArinp, OmegaCaout,
OmegaArout, HCO3inp,CO3inp, HCO3out, CO3out";
                PRINT "pHout, pCO2out, OmegaCainp, OmegaArinp, OmegaCaout,
OmegaArout, HCO3inp,CO3inp, HCO3out, CO3out";
        END SELECT
        IF MVFlag$ = "Y" THEN
                PRINT #4, ", MVFlag "
                PRINT ", MVFlag "
        ELSE
                PRINT #4, ""
                PRINT ""
        END IF
END SUB
SUB PrintInputChoices (WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$)
' SUB PrintInputChoices, version 01.01, 03-12-97, written by Ernie Lewis.
' Inputs: WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$
' Outputs: WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$
' This prints the input choices.
'
'
PrintInputModeChoice:
        PRINT "    1) Input mode: "; TAB(40);
        SELECT CASE Batch$
                CASE "NO"
                        PRINT "single-input "
                CASE "YES"
                        PRINT "batch-input  "
        END SELECT
'
'
'***********************************
PrintConstantChoice:
        PRINT "    2) Choice of constants: "; TAB(40);
        SELECT CASE WhichKs%
                CASE 1
                        PRINT "Roy et al           "
                CASE 2
                        PRINT "Goyet and Poisson   "
                CASE 3
                        PRINT "Hansson             "
                CASE 4
                        PRINT "Mehrbach et al      "
                CASE 5
                        PRINT "Dickson and Millero "
                CASE 6
                        PRINT "GEOSECS             "
                CASE 7
                        PRINT "Peng et al          "
                CASE 8
                        PRINT "freshwater          "
        END SELECT
'
'
'***********************************
PrintfCO2pCO2Choice:
        PRINT "    3) Choice of fCO2 or pCO2: "; TAB(40);
        SELECT CASE fORp$
                CASE "f"
                        PRINT "fCO2 "
                CASE "p"
                        PRINT "pCO2 "
        END SELECT
'
'
```

```
        IF WhichKs% = 8 THEN EXIT SUB
'************************************
PrintKSO4Choice:
        PRINT "    4) Choice of KSO4: "; TAB(40);
        SELECT CASE WhoseKSO4%
                CASE 1
                        PRINT "Dickson    "
                CASE 2
                        PRINT "Khoo et al "
        END SELECT
'
'
'************************************
PrintpHScaleChoice:
        PRINT "    5) Choice of pH scale: "; TAB(40);
        SELECT CASE pHScale$
                CASE "pHNBS"
                        PRINT "NBS scale      "
                CASE "pHfree"
                        PRINT "free scale     "
                CASE "pHtot"
                        PRINT "total scale    "
                CASE "pHsws"
                        PRINT "seawater scale "
                CASE "pH"
                        ' This is automatically set when WhichKs% = 8
                        ' (freshwater choice).
                        ' Only one scale is used in this case.
                        ' The tot, sws, and free scales are the same but the
                        ' NBS scale still differs because fH <> 1 due to
                        ' liquid junction potential.
        END SELECT
END SUB
SUB PrintKComments (WhichKs%)
' SUB PrintKComments, version 01.02, 05-03-97, written by Ernie Lewis.
' Inputs: WhichKs%
' Outputs: none
' This print information about the constants used in the calculations.
'
'
        PRINT
        PRINT "There will be additional uncertainties due to constants at output
conditions. "
        PRINT "K0 from Weiss, 1974. Estimates of its ACCURACY vary from .2%
to .5%. "
GetKComments:
        SELECT CASE WhichKs%
                CASE 1
                        PRINT "Roy's K1, K2; 2s PRECISION about 2% in K1, 1.5%
in K2."
                CASE 2
                        PRINT "Goyet and Poisson's K1, K2; 2s PRECISION about
2.5% in K1, 4.5% in K2. "
                CASE 3
                        PRINT "Hansson's K1, K2; 2s PRECISION about 3% in K1, 4%
in K2. "
                CASE 4
```

```
                        PRINT "Mehrbach's K1, K2; 2s PRECISION about 2.5% in K1,
4.5% in K2. "
                CASE 5
                        PRINT "Dickson and Millero's K1, K2; 2s PRECISION about
4% in K1, 6% in K2. "
                CASE 6
                        PRINT "GEOSECS choice; Mehrbach's K1, K2; 2s PRECISION
about 1.2% in K1, 2.0% in K2. "
                CASE 7
                        PRINT "Peng choice; Mehrbach's K1, K2; 2s PRECISION
about 1.2% in K1, 2.0% in K2. "
                CASE 8
                        PRINT "Freshwater; K1, K2 from Millero; 2s PRECISION
about .5% in K1, .7% in K2."
        END SELECT
END SUB
SUB PrintpHspKs (pHScale$, WhichKs%, WhoseKSO4%, TA, TC, Sal, K(), T(),
TempCinp, Pdbarinp, TempCout, Pdbarout)
' SUB PrintpHspKs, version 02.01, 10-10-97, written by Ernie Lewis.
' Inputs: pHScale$, WhichKs%, WhoseKSO4%, TA, TC, Sal, K(), T()
' Inputs:  TempCinp, Pdbarinp, TempCout, Pdbarout
' Outputs: none
' This calculates and prints the pH on all scales, and pK1, pK2, pKW, and pKB
'       on the given scale pHScale$.
'
'
FindpHsAndpKsAtInputConditions:
        TempC = TempCinp: Pdbar = Pdbarinp
        GOSUB FindpHspKs:
        pHinp = pH: fHinp = fH: pHNBSinp = pHNBS
        pHfreeinp = pHfree: pHtotinp = pHtot: pHswsinp = pHsws
        pK1inp = pK1: pK2inp = pK2: pKWinp = pKW: pKBinp = pKB
'
'
FindpHsAndpKsAtOutputConditions:
        TempC = TempCout: Pdbar = Pdbarout
        GOSUB FindpHspKs:
        pHout = pH: fHout = fH: pHNBSout = pHNBS
        pHfreeout = pHfree: pHtotout = pHtot: pHswsout = pHsws
        pK1out = pK1: pK2out = pK2: pKWout = pKW: pKBout = pKB
'
'
        S10$ = "          "
        AA2$ = "      ##.###                  ##.### "
        AA1$ = S10$ + AA2$
        IF WhichKs% = 8 THEN
                PRINT USING S10$ + "  pH  " + AA1$; pHinp; pHout
                PRINT
                PRINT USING S10$ + "  pK1 " + AA1$; pK1inp; pK1out
                PRINT USING S10$ + "  pK2 " + AA1$; pK2inp; pK2out
                PRINT USING S10$ + "  pKW " + AA1$; pKWinp; pKWout
                EXIT SUB
        END IF
'
'
        PRINT USING "       pHtot (mol/kg-SW)  " + AA2$; pHtotinp; pHtotout
        PRINT USING "       pHsws (mol/kg-SW)  " + AA2$; pHswsinp; pHswsout
        PRINT USING "       pHfree (mol/kg-SW) " + AA2$; pHfreeinp; pHfreeout
        PRINT USING "       pHNBS (mol/kg-H2O) " + AA2$; pHNBSinp; pHNBSout
        PRINT USING "       fH                 " + AA2$; fHinp; fHout
        PRINT
        PRINT
        PRINT "    These are on the "; pHScale$; : PRINT " scale ";
        SELECT CASE pHScale$
                CASE "pHtot", "pHsws", "pHfree"
                        PRINT "(mol/kg-SW): "
                CASE "pHNBS"
                        PRINT "(mol/kg-H2O): "
        END SELECT
        PRINT USING S10$ + "  pK1 " + AA1$; pK1inp; pK1out
        PRINT USING S10$ + "  pK2 " + AA1$; pK2inp; pK2out
        IF WhichKs% <> 6 THEN
                PRINT USING S10$ + "  pKW " + AA1$; pKWinp; pKWout
'               GEOSECS doesn't include OH so KW is carried as 0 in this case
        END IF
        PRINT USING S10$ + "  pKB " + AA1$; pKBinp; pKBout
EXIT SUB
'****************************************************************************
FindpHspKs:
        CALL Constants(pHScale$, WhichKs%, WhoseKSO4%, Sal, TempC, Pdbar, K0,
K(), T(), fH, FugFac, VPFac)
        K1 = K(1): K2 = K(2)
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatepHfromTATC(TA, TC, K(), T(), pH)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL FindpHOnAllScales(pHScale$, pH, K(), T(), fH, pHNBS, pHfree, pHtot,
pHsws)
        pK1 = LOG(K1) / LOG(.1)
        pK2 = LOG(K2) / LOG(.1)
        IF WhichKs% <> 8 THEN pKB = LOG(K(4)) / LOG(.1)
        IF WhichKs% <> 6 THEN pKW = LOG(K(3)) / LOG(.1)
'               GEOSECS doesn't include OH so KW is carried as 0 in this case
RETURN
END SUB
SUB PrintSpeciation (WhichKs%, TA, TC, HCO3inp, CO3inp, BAlkinp, OHinp, PAlkinp,
SiAlkinp, HCO3out, CO3out, BAlkout, OHout, PAlkout, SiAlkout)
' SUB PrintSpeciation, version 01.04, 10-10-97, written by Ernie Lewis.
' Inputs: WhichKs%, TA, TC, HCO3inp, CO3inp, BAlkinp, OHinp, PAlkinp
' Inputs: SiAlkinp, HCO3out, CO3out, BAlkout, OHout, PAlkout, SiAlkout
' Outputs: none
' This prints the contributions to the alkalinity and the speciation of the
'       carbon into HCO3, CO3, and CO2.
'
'
        AA$ = "    ####.#                  ####.# "
        S13$ = "             "
'
'
        PRINT USING S13$ + "C Alk         " + AA$; (HCO3inp + 2! * CO3inp) *
1000000!; (HCO3out + 2! * CO3out) * 1000000!
        IF WhichKs% <> 8 THEN PRINT USING S13$ + "Boron Alk     " + AA$; BAlkinp
* 1000000!; BAlkout * 1000000!
        IF WhichKs% <> 6 THEN PRINT USING S13$ + "OH            " + AA$; OHinp *
1000000!; OHout * 1000000!
```

```
        IF WhichKs% <> 8 AND WhichKs% <> 6 THEN
                PRINT USING S13$ + "Phos Alk      " + AA$; PAlkinp * 1000000!;
PAlkout * 1000000!
                PRINT USING S13$ + "Sili Alk      " + AA$; SiAlkinp * 1000000!;
SiAlkout * 1000000!
        END IF
        PRINT S13$ + S13$ + "      ------                  ------ "
        PRINT USING "        Total Alkalinity:  " + AA$; TA * 1000000!; TA *
1000000!
        PRINT
        IF WhichKs% = 6 THEN PRINT "        GEOSECS does not include OH,
phosphate, or silicate. ": PRINT
        PRINT
        PRINT USING S13$ + "HCO3-         " + AA$; HCO3inp * 1000000!; HCO3out *
1000000!
        PRINT USING S13$ + "CO3--         " + AA$; CO3inp * 1000000!; CO3out *
1000000!
        PRINT USING S13$ + "CO2*          " + AA$; (TC - HCO3inp - CO3inp) *
1000000!; (TC - HCO3out - CO3out) * 1000000!
        PRINT S13$ + S13$ + "      ------                  ------ "
        PRINT USING "        Total Inorganic C: " + AA$; TC * 1000000!; TC *
1000000!
END SUB
SUB PrintTCfCO2Warning
' SUB PrintTCfCO2Warning, version 01.02, 10-10-97, written by Ernie Lewis.
' Inputs: none
' Outputs: none
' This prints a note in the case of the combination TC, fCO2 being
'       either physically impossible or so close that it becomes so
'       during calculations to determine sensitivity to inputs.
'
'
        PRINT
        PRINT "    Since TC = HCO3 + CO3 + (fCO2 * K0), (fCO2 * K0) must be <
TC. "
        PRINT "    For the combination of TC and fCO2 (or pCO2) entered this
condition "
        PRINT "    is violated or else is so close that it becomes so when
calculating "
        PRINT "    sensitivity to input conditions. "
        PRINT
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer.
        INPUT "    Please retry - hit <enter> to continue. ", Q$
END SUB
SUB RevelleFactor (WhichKs%, TA, TC, K0, K(), T(), Revelle)
' SUB RevelleFactor, version 01.03, 01-07-97, written by Ernie Lewis.
' Inputs: WhichKs%, TA, TC, K0, K(), T()
' Outputs: Revelle
' This calculates the Revelle factor (dfCO2/dTC)|TA/(fCO2/TC).
' It only makes sense to talk about it at pTot = 1 atm, but it is computed
'       here at the given K(), which may be at pressure <> 1 atm. Care must
'       thus be used to see if there is any validity to the number computed.
'
'
        IF TC = 0! THEN Revelle = 0!: EXIT SUB
        K1 = K(1): K2 = K(2)
        TC0 = TC
        dTC = .000001: ' 1 umol/kg-SW
'
'
' Find fCO2 at TA, TC + dTC
        TC = TC0 + dTC
        GOSUB GetfCO2:
        fCO2plus = fCO2
'
'
' Find fCO2 at TA, TC - dTC
        TC = TC0 - dTC
        GOSUB GetfCO2:
        fCO2minus = fCO2
'
CalculateRevelleFactor:
        Revelle = (fCO2plus - fCO2minus) / dTC / ((fCO2plus + fCO2minus) / TC)
        ' at constant TA
'
'
ResetTC:
        TC = TC0
EXIT SUB
'****************************************************************************
GetfCO2:
        IF WhichKs% = 7 THEN TA = TA - T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatepHfromTATC(TA, TC, K(), T(), pH)
        IF WhichKs% = 7 THEN TA = TA + T(4): ' PAlk(Peng) = PAlk(Dickson) + TP
        CALL CalculatefCO2fromTCpH(TC, pH, K0, K1, K2, fCO2)
RETURN
END SUB
SUB SetDefaultsForCO2SYS (ICase%, WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$,
TA, TC, pHinp, fCO2inp, pCO2inp, Sal, TempCinp, Pdbarinp, TempCout, Pdbarout,
NHeaderLines%, NIDFields%, MVD, MVFlag$)
' SUB SetDefaultsForCO2SYS, version 03.04, 10-10-97, written by Ernie Lewis.
' Inputs: none
' Outputs: ICase%, WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$
' Outputs: TA, TC, pHinp, fCO2inp, pCO2inp, Sal, TempCinp, Pdbarinp
' Outputs: TempCout, Pdbarout, NHeaderLines%, NIDFields%, MVD, MVFlag$
' This sets default values. They may be changed later in the program.
'
'
        ICase% = 1:             ' given TA, TC find others
        WhichKs% = 1:           ' Roy's K1, K2
        WhoseKSO4% = 1          ' Dickson's KSO4
        fORp$ = "f":            ' fCO2
        pHScale$ = "pHtot":     ' Total pH scale
        Batch$ = "NO":          ' single-input mode
        TA = .0023:             ' mol/kg-SW
        TC = .0021:             ' mol/kg-SW
        pHinp = 7.9:            '
        fCO2inp = .0006:        ' atm
        pCO2inp = .0006:        ' atm
        Sal = 35!:              ' mille
        TempCinp = 20!:         ' deg C
        Pdbarinp = 0!:          ' decibars
        TempCout = 5!:          ' deg C
        Pdbarout = 0!:          ' decibars
```

```
' for batch-input mode:
        NHeaderLines% = 1:      ' number of header lines in input file
        NIDFields% = 1:         ' number of ID fields per sample
        MVD = -9:               ' missing value designator
        MVFlag$ = "Y":          ' missing value flag
END SUB
SUB SetInputChoices (WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$)
' SUB SetInputChoices, version 01.02, 03-11-97, written by Ernie Lewis.
' Inputs: WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$
' Outputs: WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$
' This allows the user to change the default settings.
'
'
TopOfChooseHere:
        IF WhichKs% <> 8 AND pHScale$ = "pH" THEN pHScale$ = "pHtot"
        ' this is in case it was that way and then changed later
        IF WhichKs% = 8 THEN
                WhoseKSO4% = 1: ' just so there is always a default value
                pHScale$ = "pH": ' the scales should be the same here
        END IF
'
'
        CLS
        PRINT "    Enter number to change, <enter> to continue, or <E> to end: "
        PRINT
        CALL PrintInputChoices(WhichKs%, WhoseKSO4%, fORp$, pHScale$, Batch$)
        PRINT
        PRINT "    Enter <I> for information about this program. "
'
'
'***********************************
ChangeInputChoices:
        DO: LOOP WHILE INKEY$ <> "": 'This clears the key buffer
        LOCATE 1, 65: INPUT "", Q$
        IF Q$ = "E" OR Q$ = "e" THEN CLS : END
        IF Q$ = "I" OR Q$ = "i" THEN
                Info$ = "Menu": CALL AboutCO2SYS(Info$)
                GOTO TopOfChooseHere:
        ELSE
                Q% = VAL(Q$)
        END IF
        SELECT CASE Q%
                CASE 0
                        LOCATE 1, 1: PRINT SPACE$(80): PRINT SPACE$(80)
                        EXIT SUB
                CASE 1
                        CALL ChooseInputMode(Batch$)
                CASE 2
                        CALL ChooseWhichKs(WhichKs%, pHScale$)
                CASE 3
                        CALL ChoosefORp(fORp$)
                CASE 4
                        IF WhichKs% <> 8 THEN CALL ChooseWhoseKSO4(WhoseKSO4%)
                CASE 5
                        IF WhichKs% <> 8 THEN CALL ChoosepHScale(pHScale$)
                CASE ELSE
        END SELECT
        GOTO TopOfChooseHere:
END SUB
SUB SetParametersForPartials (dTA, dTC, dpH, dfCO2, dSal, dTempC, dPdbar, pcdK0,
pcdK1, pcdK2)
' SUB SetParametersForPartials, version 01.00, 11-26-96, written by Ernie Lewis.
' Inputs: none
' Outputs: dTA, dTC, dpH, dfCO2, dSal, dTempC, dPdbar, pcdK0, pcdK1, pcdK2
' This sets the changes used for finding partials. Since they are found
'       numerically there will be some roundoff error, but the values
'       below seem to work ok.
'
'
        dTA = .000001:          ' 1 umol/kg-SW
        dTC = .000001:          ' 1 umol/kg-SW
        dpH = .001:             ' 1 milli-pH
        dfCO2 = .000001:        ' 1 uatm
        dSal = 1!:              ' 1 mille
        dTempC = 1!:            ' 1 deg C
        dPdbar = 1!:            ' 1 dbar
        pcdK0 = 1!:             ' 1% change in K0
        pcdK1 = 1!:             ' 1% change in K1
        pcdK2 = 1!:             ' 1% change in K2
END SUB
```