FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '90
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will display the initials of NCNB.
'
PRINT "NN    N  CCCCC  NN    N  BBBB"
PRINT "N N   N  C      N N   N  B   B"
PRINT "N  N  N  C      N  N  N  BBBBB"
PRINT "N   N N  C      N   N N  B   B"
PRINT "N    NN  CCCCC  N    NN  BBBB"
```

```
'1.2
' This program will print the name of the SYSTEM.
'
INPUT "Enter #:"; N
PRINT "SYSTEM"; N
```

```
'1.3
' This program will display the value of programmers.
'
INPUT "Enter N: "; N
PRINT 66 + N; "BILLION DOLLARS"
```

```
'1.4
' This program will indicate the county for zip code.
'
INPUT "Enter zip code: "; N
IF N = 33701! OR N = 34685! OR N = 34646! THEN
  PRINT "PINELLAS"
ELSE
  IF N = 33525! OR N = 34249! OR N = 34690! THEN
    PRINT "PASCO"
  ELSE
    PRINT "HILLSBOROUGH"
  END IF
END IF
```

```
'1.5
' This program will display Hugh McColl's goals.
'
INPUT "Enter MMM: "; M
INPUT "Enter YYYY: "; Y
PRINT "HUGH MCCOLL WOULD LIKE NCNB TO GROW"
PRINT "TO"; M; "BILLION DOLLARS IN ASSETS BY"
PRINT "THE YEAR"; Y
```

```
'1.6
' This program will calculate maximum number of coupons.
'
INPUT "Enter N associates: "; N
INPUT "Enter C coupons: "; C
PRINT INT(C / N + .99)
```

```
'1.7
' This program will print divisions in COBOL program.
'
INPUT "Enter division: "; D$
SELECT CASE D$
  CASE "IDENTIFICATION"
    PRINT "BEFORE = NONE"
    PRINT "AFTER = ENVIRONMENT  DATA  PROCEDURE"
  CASE "ENVIRONMENT"
    PRINT "BEFORE = IDENTIFICATION"
    PRINT "AFTER = DATA  PROCEDURE"
  CASE "DATA"
    PRINT "BEFORE = IDENTIFICATION  ENVIRONMENT"
    PRINT "AFTER = PROCEDURE"
  CASE "PROCEDURE"
    PRINT "BEFORE = IDENTIFICATION  DATA  PROCEDURE"
    PRINT "AFTER = NONE"
END SELECT
```

```
'1.8
' This program will display states having holidays.
'
INPUT "Enter N: "; N
IF N <= 7 THEN PRINT "FL NC SC TX MD GA VA": END
IF N = 8 THEN PRINT "FL NC TX MD GA VA": END
IF N = 9 OR N = 10 THEN PRINT "FL TX MD GA VA": END
IF N = 11 THEN PRINT "MD"
```

```
'1.9
' This program will correct modern dates.
'
INPUT "Enter date: "; D
INPUT "Enter A.D. or B.C.: "; A$
IF A$ = "B.C." AND D > 4 THEN PRINT D - 4; "B.C.": END
IF A$ = "B.C." THEN PRINT 5 - D; "A.D.": END
PRINT D + 4; "A.D"
```

```
'1.10
' This program will print a 7 letter word diamond.
'
INPUT "Enter word: "; N$
PRINT "   "; MID$(N$, 4, 1)
PRINT "  "; MID$(N$, 3, 3)
PRINT " "; MID$(N$, 2, 5)
PRINT MID$(N$, 1, 7)
PRINT " "; MID$(N$, 2, 5)
PRINT "  "; MID$(N$, 3, 3)
PRINT "   "; MID$(N$, 4, 1)
```

```
'2.1
' This program will encode a phrase.
'
INPUT "Enter phrase: "; A$
FOR I = 1 TO LEN(A$)
  C$ = MID$(A$, I, 1)
  IF C$ < "A" OR C$ > "Z" THEN
    PRINT C$;
  ELSE
    IF C$ = "A" THEN
      PRINT "Z";
    ELSE
      PRINT CHR$(ASC(C$) - 1);
    END IF
  END IF
NEXT I
```

```
'2.2
' This program will determine the "type" of year.
'
INPUT "Enter year: "; Y
IF Y / 10 = INT(Y / 10) THEN PRINT "END OF DECADE"
IF Y / 100 = INT(Y / 100) THEN PRINT "END OF CENTURY"
IF Y / 1000 = INT(Y / 1000) THEN PRINT "END OF MILLENNIUM"
IF Y - INT(Y / 10) * 10 = 1 THEN PRINT "BEGINNING OF DECADE"
IF Y - INT(Y / 100) * 100 = 1 THEN PRINT "BEGINNING OF CENTURY"
IF Y - INT(Y / 1000) * 1000 = 1 THEN
  PRINT "BEGINNING OF MILLENNIUM"
END IF
```

```
'2.3
' This program will print average and handicap of bowlers.
'
A$(1) = "BOB:   ": A$(2) = "DOUG:  ": A$(3) = "JACKIE:"
A$(4) = "JOSE:  "
FOR I = 1 TO 4
  PRINT "Enter scores for "; A$(I); : INPUT S1, S2, S3
  AVE(I) = (S1 + S2 + S3) / 3
  IF AVE(I) > 200 THEN
    HAN(I) = 0
  ELSE
    HAN(I) = (200 - AVE(I)) * .9
  END IF
NEXT I
FOR I = 1 TO 4
  PRINT A$(I);
  PRINT USING " AVERAGE = ###"; INT(AVE(I) + .01);
  PRINT "  HANDICAP ="; INT(HAN(I) + .01)
NEXT I
```

```
'2.4
' This program will determine # of days to add to date.
'
INPUT "Enter date: "; D$
MM = VAL(LEFT$(D$, 2))
DD = VAL(MID$(D$, 4, 2))
YY = VAL(RIGHT$(D$, 4))
PRINT "ADD ";
IF YY < 1700 OR (YY = 1700 AND MM < 3) THEN PRINT "10 DAYS": END
IF YY < 1800 OR (YY = 1800 AND MM < 3) THEN PRINT "11 DAYS": END
IF YY < 1900 OR (YY = 1900 AND MM < 3) THEN PRINT "12 DAYS": END
IF YY < 2100 OR (YY = 2100 AND MM < 3) THEN PRINT "13 DAYS": END
```

```
'2.5
' This program will sort efficiencies of sorting algorithms.
'
N$(1) = "BUBBLE SORT": N$(2) = "SHELL SORT": N$(3) = "QUICK SORT"
INPUT "Enter N: "; N
A(1) = N * (N - 1) / 2
A(2) = N * (LOG(N) / LOG(2)) * (LOG(N) / LOG(2))
A(3) = N * (LOG(N) / LOG(2))
FOR I = 1 TO 2
  FOR J = I + 1 TO 3
    IF A(I) > A(J) THEN
      SWAP A(I), A(J): SWAP N$(I), N$(J)
    END IF
  NEXT J
NEXT I
FOR I = 1 TO 3: PRINT N$(I): NEXT I
```

```
'2.6
' This program will determine status for each hole of golf.
'
DATA 4, 3, 4, 5, 4, 3, 5, 4, 4
FOR I = 1 TO 9: READ P(I): PAR = PAR + P(I): NEXT I
FOR I = 1 TO 9
  PRINT "Enter score for hole"; I; : INPUT S(I)
  SUM = SUM + S(I)
NEXT I
PRINT "HOLE  PAR  SCORE  STATUS"
PRINT "----  ---  -----  ------"
FOR I = 1 TO 9
  PRINT I; "   "; P(I); "   "; S(I); "   ";
  D = S(I) - P(I)
  SELECT CASE D
    CASE -3: PRINT "DOUBLE EAGLE"
    CASE -2: PRINT "EAGLE"
    CASE -1: PRINT "BIRDIE"
    CASE 0: PRINT "PAR"
    CASE 1: PRINT "BOGEY"
    CASE 2: PRINT "DOUBLE BOGEY"
  END SELECT
NEXT I
PRINT "      ---  -----"
PRINT "     "; PAR; "  "; SUM
```

```
'2.7
' This program will determine time calendar is ahead/behind.
'
INPUT "Enter N: "; N
'   Sum 5 hours 48 min 47.8 sec for every year
H = 5 * N: M = 48 * N: S = 47.8 * N
'   Convert to standard form
SN = INT(S / 60): S = S - SN * 60: M = M + SN
MN = INT(M / 60): M = M - MN * 60: H = H + MN
HN = INT(H / 24): H = H - HN * 24: D = HN
'   Subtract 1 for every leap year counted
LY = INT(N / 4)
IF LY <= D THEN
  PRINT D - LY; "DAYS "; H; "HOURS "; M; "MIN  ";
  PRINT USING "##.# SEC  AHEAD"; S
ELSE
  PRINT (LY - D - 1); "DAYS ";
  PRINT 23 - H; "HOURS "; 59 - M; "MIN  ";
  PRINT USING "##.# SEC  BEHIND"; 60 - S
END IF
```

```
'2.8
' This program will display members on a committee.
'
DATA JACKIE,TOM,LOVETTA,GREG,TONY,AL,KAREN
DATA JAN,NORM,TRUDY,THERESA,ALICE,DAVE,JIM,STEVE
DIM A$(20)
FOR I = 1 TO 15: READ A$(I): NEXT I
N$(1) = "BARB": NM(1) = 6: N$(2) = "JOE": NM(2) = 8
N$(3) = "DOUG": NM(3) = 9: Y = 1989: M = 9
INPUT "Enter month, year: "; MONTH, YEAR
PRINT USING "##/"; M; : PRINT USING "#### - "; Y;
PRINT N$(1); "  "; N$(2); "  "; N$(3)
I = 1
WHILE (M <> MONTH) OR (Y <> YEAR)
  M = M + 1: IF M = 13 THEN M = 1: Y = Y + 1
  FOR J = 1 TO 3
    IF ABS(M - NM(J)) = 6 THEN
      N$(J) = A$(I): I = I + 1: NM(J) = M
      PRINT USING "##/"; M; : PRINT USING "#### - "; Y;
      PRINT N$(1); "  "; N$(2); "  "; N$(3)
    END IF
  NEXT J
WEND
```

```
'2.9
' This program will graph the sine and cosine functions.
'
FOR F = 1 TO 2
  CLS
  FOR I = 1 TO 24: LOCATE I, 40: PRINT "!"; : NEXT I
  LOCATE 12, 1
  FOR I = 1 TO 79: PRINT "-"; : NEXT I
  LOCATE 12, 40: PRINT "+";
  CINC = 39 / 3.14: RINC = 11
  FOR X = -3.14 TO 3.14 STEP .05
     C = 40 + CINC * X
     IF F = 1 THEN R = 12 - SIN(X) * RINC
     IF F = 2 THEN R = 12 - COS(X) * RINC
     LOCATE R, C: PRINT "*";
   NEXT X
   A$ = "": WHILE A$ = "": A$ = INKEY$: WEND
 NEXT F
CLS
```

```
'2.10
' This program will estimate hours of training given choices.
'
CLS
PRINT "           NCNB IN-HOUSE TRAINING LIST"
PRINT
PRINT "COURSE #   COURSE NAME               EST. HOURS"
PRINT "--------   -----------               ----------"
A$(1) = "187-11X": B$(1) = "ISPF/PDS FUNDAMENTALS     6.5 - 8"
A$(2) = "187-15X": B$(2) = "ISPF/PDS FOR PROGRAMMERS  4.5 - 6"
A$(3) = "220-AXX": B$(3) = "JCL FUNDAMENTALS           15 - 20"
A$(4) = "200-AXX": B$(4) = "VSAM CONCEPTS               4 - 7"
A$(5) = "123-2XX": B$(5) = "MVS/SP/XA VSAM              7 - 11"
A$(6) = "130-11X": B$(6) = "CICS/VS SKILLS I            6 - 8"
A$(7) = "130-15X": B$(7) = "CICS/VS SKILLS II           4 - 6"
DATA 6.5,8, 4.5,6, 15,20, 4,7, 7,11, 6,8, 4,6
FOR I = 1 TO 7: READ LOW(I), HIGH(I): NEXT I
FOR I = 1 TO 7
  PRINT A$(I); "    "; B$(I)
NEXT I
PRINT : NUM = 0
INPUT "Enter course # (or 000-000 to end): "; C$
WHILE C$ <> "000-000"
  I = 1: WHILE C$ <> A$(I): I = I + 1: WEND
  NUM = NUM + 1: C(NUM) = I
  LSUM = LSUM + LOW(I): HSUM = HSUM + HIGH(I)
  INPUT "Enter course # (or 000-000 to end): "; C$
WEND
'    Display options selected and TOTAL estimated hours
CLS
PRINT "COURSE NAME               EST. HOURS"
PRINT "-----------               ----------"
FOR I = 1 TO NUM: PRINT B$(C(I)): NEXT I
PRINT "                          ----------"
PRINT USING "                 TOTAL = ##.#"; LSUM;
PRINT " -"; HSUM; "HOURS"
```

```
'3.1
' This program will produce acronyms for phone numbers.
'
DIM A$(18)
DATA AGENT,SOAP,MONEY,JEWEL,BALL,LOANS,CARE,SAVE,CALL
DATA PAVE,KEEP,KINGS,KNIFE,KNOCK,JOINT,JUICE,LOBBY,RATE
FOR I = 1 TO 18: READ A$(I): NEXT I
DATA A,B,C, D,E,F, G,H,I, J,K,L, M,N,O, P,R,S, T,U,V, W,X,Y
FOR I = 2 TO 9: READ L1$(I), L2$(I), L3$(I): NEXT I
INPUT "Enter phone #: "; PH$
P4$ = RIGHT$(PH$, 4): P5$ = MID$(PH$, 3, 1) + P4$
'   Convert words to number strings
FOR I = 1 TO 18
  L = LEN(A$(I)): NUM$ = ""
  FOR J = 1 TO L
    K = 2: C$ = MID$(A$(I), J, 1)
    WHILE (L1$(K) <> C$) AND (L2$(K) <> C$) AND (L3$(K) <> C$)
      K = K + 1
    WEND
    NUM$ = NUM$ + LTRIM$(STR$(K))
  NEXT J
  IF L = 4 AND NUM$ = P4$ THEN
    PRINT LEFT$(PH$, 4); A$(I)
  ELSE
    IF L = 5 AND NUM$ = P5$ THEN
      PRINT LEFT$(PH$, 2); LEFT$(A$(I), 1); "-";
      PRINT RIGHT$(A$(I), 4)
    END IF
  END IF
NEXT I
```

```
'3.2
' This program will select words given a string w/ wildcard.
'
DATA COMPUTE, COMPUTER, COMPUTERS, COMPORT, COMPUTES
DATA COMPUTED, ATTRACTIVE, ABRASIVE, ADAPTIVE, ACCEPTIVE
DATA AERATING, CONTESTED, CONTESTER, CORONETS, CONTESTS
DATA CONTESTERS, COUNTESS, CREATIVE, CREATE, CREATURE
DATA CREATION, EVERYBODY, EVERYONE, EMPTY, ELECTION
DIM A$(25)
N = 25: FOR I = 1 TO N: READ A$(I): NEXT I: I = 0
DO UNTIL I > L
  INPUT "Enter string: "; A$: L = LEN(A$): W = 0: I = 0: X$ = ""
  WHILE (I <= L) AND (X$ <> "*")
    I = I + 1: X$ = MID$(A$, I, 1)
  WEND
  IF I > L THEN END
'     Asterisk is position I
  L$ = LEFT$(A$, I - 1): R$ = RIGHT$(A$, L - I)
  FOR J = 1 TO N
    IF LEFT$(A$(J), I - 1) = L$ AND RIGHT$(A$(J), L - I) = R$ THEN
      PRINT A$(J); "  "; : W = 1
    END IF
  NEXT J
  IF W = 0 THEN PRINT "NO WORDS FOUND"
  PRINT
LOOP
```

```
'3.3
' This program will keep score for a double dual race.
'
CLS : DIM IN$(21)
FOR I = 1 TO 21
  PRINT "Place "; I; ":"; : INPUT IN$(I)
  IF I > 1 THEN
    J = 1
    WHILE J <= TN AND INIT$(J) <> IN$(I): J = J + 1: WEND
  END IF
  IF (INIT$(J) <> IN$(I)) OR (I = 1) THEN
    TN = TN + 1: INIT$(TN) = IN$(I)
  END IF
NEXT I
'    Assert  TEAM$(1, 2, 3) = 3 unique team INITIALS
FOR I = 1 TO 2
  FOR J = I + 1 TO 3
    PL = 0: T1 = 0: T2 = 0: T1PL = 0: T2PL = 0
    FOR K = 1 TO 21
      IF IN$(K) = INIT$(I) THEN
     PL = PL + 1: T1 = T1 + PL: T1PL = T1PL + 1
     TEAM1(T1PL) = PL
      END IF
      IF IN$(K) = INIT$(J) THEN
     PL = PL + 1: T2 = T2 + PL: T2PL = T2PL + 1
     TEAM2(T2PL) = PL
      END IF
    NEXT K
    T1 = T1 - TEAM1(6) - TEAM1(7)
    T2 = T2 - TEAM2(6) - TEAM2(7)
    PRINT "TEAM "; INIT$(I); ":"; T1; " POINTS"
    PRINT "TEAM "; INIT$(J); ":"; T2; " POINTS"
    IF (T1 < T2) OR (T1 = T2 AND TEAM1(6) < TEAM2(6)) THEN
      PRINT "TEAM "; INIT$(I);
    ELSE
      PRINT "TEAM "; INIT$(J);
    END IF
    PRINT " WINS!": PRINT
  NEXT J
NEXT I
```

```
'3.4
' This program will determine who gets which program #s.
'
DEFINT A-Z
INPUT "Enter X, Y, Z: "; X, Y, Z
A$(1) = "AL, DOUG, AND JAN = "
A$(2) = "AL AND DOUG = "
A$(3) = "AL AND JAN = "
A$(4) = "DOUG AND JAN = "
A$(5) = "AL = "
A$(6) = "DOUG = "
A$(7) = "JAN = "
A$(8) = "NORM = "
FOR K = 1 TO 8
  PRINT A$(K); : ONE = 0
  FOR I = 1 TO 30
    XD = (I / X = INT(I / X)): YD = (I / Y = INT(I / Y))
    ZD = (I / Z = INT(I / Z))
    I$ = LTRIM$(STR$(I)) + " "
    IF K = 1 AND XD AND YD AND ZD THEN PRINT I$; : ONE = 1
    IF K = 2 AND XD AND YD AND NOT ZD THEN PRINT I$; : ONE = 1
    IF K = 3 AND XD AND NOT YD AND ZD THEN PRINT I$; : ONE = 1
    IF K = 4 AND NOT XD AND YD AND ZD THEN PRINT I$; : ONE = 1
    IF K = 5 AND XD AND NOT YD AND NOT ZD THEN PRINT I$; : ONE = 1
    IF K = 6 AND NOT XD AND YD AND NOT ZD THEN PRINT I$; : ONE = 1
    IF K = 7 AND NOT XD AND NOT YD AND ZD THEN PRINT I$; : ONE = 1
    IF K = 8 AND NOT XD AND NOT YD AND NOT ZD THEN
      PRINT I$; : ONE = 1
    END IF
  NEXT I
  IF ONE = 0 THEN PRINT "NONE" ELSE PRINT
NEXT K
```

```
'3.5
' This program will display numbers 1-8 and a blank in a
' 3 x 3 array.  When a digit is pressed, it moves into the
' blank (if possible).
'
RANDOMIZE TIMER
'  Assign numbers in array sequentially then scramble them
FOR I = 1 TO 3
  FOR J = 1 TO 3
    A(I, J) = (I - 1) * 3 + J - 1
  NEXT J
NEXT I
FOR I = 1 TO 3
  FOR J = 1 TO 3
    R1 = INT(RND(3) * 3) + 1: R2 = INT(RND(3) * 3) + 1
    X = A(I, J): A(I, J) = A(R1, R2): A(R1, R2) = X
  NEXT J
NEXT I
'
WHILE (DIG <> 9)
'     Display Array
  CLS
  FOR I = 1 TO 3
    FOR J = 1 TO 3
      IF A(I, J) > 0 THEN PRINT A(I, J); " ";
      IF A(I, J) = 0 THEN PRINT "    "; : BX = I: BY = J
    NEXT J: PRINT
  NEXT I
'     Accept valid digit or 9 (to end)
  VALID = 0
  WHILE (VALID = 0) AND (DIG <> 9)
    A$ = "": WHILE A$ = "": A$ = INKEY$: WEND
    DIG = VAL(A$)
    FOR I = 1 TO 3
      FOR J = 1 TO 3
     IF DIG = A(I, J) THEN IX = I: IY = J
      NEXT J
    NEXT I
    IF ABS(BX - IX) + ABS(BY - IY) = 1 THEN VALID = -1
  WEND
'
  IF VALID THEN
'     Move digit into blank space
    X = A(IX, IY): A(IX, IY) = A(BX, BY): A(BX, BY) = X
  END IF
WEND
```

```
'3.6
' This program will simulate the moves of a chess game.
'
A$(8) = "BR1 BK1 BB1 BQ  BK  BB2 BK2 BR2   !  8"
A$(7) = "BP1 BP2 BP3 BP4 BP5 BP6 BP7 BP8   !  7"
A$(6) = "                                  !  6"
A$(5) = "                                  !  5"
A$(4) = "                                  !  4"
A$(3) = "                                  !  3"
A$(2) = "WP1 WP2 WP3 WP4 WP5 WP6 WP7 WP8   !  2"
A$(1) = "WR1 WK1 WB1 WQ  WK  WB2 WK2 WR2   !  1"
A$(9) = "-----------------------------------"
A$(10) = " A   B   C   D   E   F   G   H"
CLS : L = LEN(A$(1))
FOR I = 8 TO 1 STEP -1: PRINT A$(I): NEXT I
PRINT A$(9): PRINT A$(10)
WKR = 1: WKC = 5: BKR = 8: BKC = 5   'Location of 2 kings
'
WHILE (R2 <> WKR OR C2 <> WKC) AND (R2 <> BKR OR C2 <> BKC)
  LOCATE 12, 1: PRINT SPACE$(30): LOCATE 12, 1
  IF MOV = 0 THEN INPUT "Enter white move: "; M$
  IF MOV = 1 THEN INPUT "Enter black move: "; M$
'     Convert moves to coordinates
  C1 = ASC(LEFT$(M$, 1)) - 64: R1 = VAL(MID$(M$, 2, 1))
  C2 = ASC(MID$(M$, 4, 1)) - 64: R2 = VAL(RIGHT$(M$, 1))
'     Move piece from 1 string to another and redisplay
  PIEC$ = MID$(A$(R1), (C1 - 1) * 4 + 1, 4)
  L$ = LEFT$(A$(R2), (C2 - 1) * 4)
  R$ = RIGHT$(A$(R2), L - C2 * 4)
  A$(R2) = L$ + PIEC$ + R$
  LOCATE 9 - R2, 1: PRINT A$(R2)
'     Remove piece from string by placing spaces and redisplay
  L$ = LEFT$(A$(R1), (C1 - 1) * 4)
  R$ = RIGHT$(A$(R1), L - C1 * 4)
  A$(R1) = L$ + "    " + R$
  LOCATE 9 - R1, 1: PRINT A$(R1)
'     If a king moved, store new location
  IF R1 = WKR AND C1 = WKC THEN WKR = R2: WKC = C2: R2 = 0: C2 = 0
  IF R1 = BKR AND C1 = BKC THEN BKR = R2: BKC = C2: R2 = 0: C2 = 0
  IF MOV = 0 THEN MOV = 1 ELSE MOV = 0
WEND
LOCATE 12, 1: PRINT "CHECK MATE, ";
IF R2 = WKR AND C2 = WKC THEN PRINT "BLACK WON     ": END
PRINT "WHITE WON     "
```

```
'3.7
' This program will print date of Easter and Lent in a year.
'
DATA 4,14, 4,3, 3,23, 4,11, 3,31, 4,18, 4,8, 3,28, 4,16, 4,5
DATA 3,25, 4,13, 4,2, 3,22, 4,10, 3,30, 4,17, 4,7, 3,27
DIM M(18), D(18)
FOR I = 0 TO 18: READ M(I), D(I): NEXT I
MD(1) = 31: MD(2) = 28: MD(3) = 31
MO$(2) = "FEBRUARY":  MO$(3) = "MARCH":  MO$(4) = "APRIL"
INPUT "Enter year: "; Y
KE = Y - INT(Y / 19) * 19
' Calculate # of days between 1,1,1970 and date
DAYS = (Y - 1970) * 365 + INT((Y - 1968) / 4)
FOR I = 1 TO M(KE) - 1:  DAYS = DAYS + MD(I): NEXT I
DAYS = DAYS + D(KE)
X = DAYS - INT(DAYS / 7) * 7
' if X = 0-Wed, 1-Thu, 2-Fri, 3-Sat, 4-Sun, 5-Mon, 6-Tue
IF X = 0 OR X = 1 OR X = 2 OR X = 3 THEN EDAY = D(KE) + (4 - X)
IF X = 4 OR X = 5 OR X = 6 THEN EDAY = D(KE) + (11 - X)
EMON = M(KE)
IF M(KE) = 3 AND EDAY > MD(3) THEN
  EDAY = EDAY - MD(3): EMON = EMON + 1
END IF
PRINT "EASTER IS ON "; MO$(EMON); EDAY
'  Compute date of Lent
LMON = EMON - 1: LDAY = MD(LMON) + EDAY - 46
IF LDAY < 1 THEN LMON = LMON - 1:  LDAY = LDAY + MD(LMON)
IF LMON = 2 AND Y / 4 = INT(Y / 4) THEN LDAY = LDAY + 1
PRINT "LENT IS ON "; MO$(LMON); LDAY
```

```
'3.8
' This program will keep score for a bowler.
'
DIM A(10, 3): WIDTH 40: CLS
FOR I = 1 TO 10: PRINT "Enter frame"; I; : INPUT A$(I): NEXT I
PRINT
PRINT "-1- -2- -3- -4- -5- -6- -7- -8- -9- -10-";
PRINT "---!---!---!---!---!---!---!---!---!---!";
FOR I = 1 TO 10
  PRINT SPACE$(3 - LEN(A$(I))); A$(I); "!";
NEXT I
'
' Assign values to A FRames according to X, /, or pins
'
FOR FR = 1 TO 10
  L = LEN(A$(FR))
  FOR J = 1 TO L
    MD$ = MID$(A$(FR), J, 1)
    IF MD$ = "X" THEN
      A(FR, J) = 10: LOOK(FR) = 2
    ELSE
      IF MD$ = "/" THEN
        A(FR, J) = 10 - A(FR, J - 1): LOOK(FR) = 1
      ELSE
        A(FR, J) = VAL(MD$)
      END IF
    END IF
  NEXT J
NEXT FR
'
' Determine FRame values with LOOK ahead
'
FOR FR = 1 TO 10
  SUM(FR) = SUM(FR - 1) + A(FR, 1) + A(FR, 2)
  IF LOOK(FR) > 0 THEN
    IF LOOK(FR) <= 1 THEN
'      ***  A spare / needs 1 more value added  ***
        IF FR = 10 THEN
          SUM(FR) = SUM(FR) + A(FR, 3)
        ELSE
          SUM(FR) = SUM(FR) + A(FR + 1, 1)
        END IF
    ELSE
 '      ***  A strike X needs 2 more values added  ***
      IF FR = 10 THEN
        SUM(FR) = SUM(FR) + A(FR, 3)
      ELSE
        SUM(FR) = SUM(FR) + A(FR + 1, 1) + A(FR + 1, 2)
        IF FR <> 9 THEN
          IF A(FR + 1, 1) = 10 THEN
            SUM(FR) = SUM(FR) + A(FR + 2, 1)
          END IF
        END IF
      END IF
    END IF
  END IF
 '      *** Print FRame's value ***
  SUM$ = MID$(STR$(SUM(FR)), 2)
  PRINT SUM$; SPACE$(3 - LEN(SUM$)); "!";
NEXT FR
PRINT STRING$(40, "-")
```

```
'3.9
' This program will solve an N x N system of equations.
'
INPUT "Enter N: "; N
FOR ROW = 1 TO N
  PRINT "Enter coefficients for row"; ROW
  FOR COL = 1 TO N
    PRINT "CO"; COL; ": "; : INPUT C(ROW, COL)
  NEXT COL
  INPUT "Enter constant: "; C(ROW, N + 1)
NEXT ROW
'     Make main diagonals all 1s with 0s to the left
FOR ROW = 1 TO N
  DEN = C(ROW, ROW)
  FOR COL = ROW TO N + 1
    C(ROW, COL) = C(ROW, COL) / DEN
  NEXT COL
  FOR R = ROW + 1 TO N
    X = C(R, ROW)
    FOR COL = ROW TO N + 1
      C(R, COL) = C(R, COL) - X * C(ROW, COL)
    NEXT COL
  NEXT R
NEXT ROW
'    Make 0s on the right of 1s on main diagonal, not const
FOR ROW = N TO 1 STEP -1
  FOR R = ROW - 1 TO 1 STEP -1
    X = C(R, ROW)
    FOR COL = ROW TO N + 1
      C(R, COL) = C(R, COL) - X * C(ROW, COL)
    NEXT COL
  NEXT R
NEXT ROW
'   Display solution
PRINT "("; LTRIM$(STR$(INT(C(1, N + 1) + .1)));
FOR ROW = 2 TO N: PRINT ", ";
  PRINT LTRIM$(STR$(INT(C(ROW, N + 1) + .1)));
NEXT ROW
PRINT ")"
```

```
'3.10
' This program will solve crytorithms with two 2-letter addends
' and a 3-letter sum, using only the letters A, B, C, D, and E.
'
DEFINT A-Z
INPUT "Enter first addend: "; S1$
INPUT "Enter second addend: "; S2$
INPUT "Enter sum: "; S3$
L$ = S1$ + S2$ + S3$
'  Store in FL() the index of the first occurence
FOR I = 1 TO 7
  CH$ = MID$(L$, I, 1)
  J = 1: WHILE MID$(L$, J, 1) <> CH$: J = J + 1: WEND
  FL(I) = J
   IF J = I THEN NL = NL + 1: UL(NL) = I  'A new letter
NEXT I
'
FOR N1 = 10 TO 98           'N1 must be 2 digits, >9
  FOR N2 = 100 - N1 TO 98   'N2 must be 2 digits, >9
    SUM = N1 + N2           'Sum must be 3 digits >99
    N1$ = LTRIM$(STR$(N1))
    N2$ = LTRIM$(STR$(N2))
    SUM$ = LTRIM$(STR$(SUM))
    NS$ = N1$ + N2$ + SUM$
    I = 1:  SOL = 1
'      Check if similar letters correspond to similar #s
    WHILE (I <= 7) AND (SOL = 1)
      CH$ = MID$(NS$, I, 1)
      IF CH$ <> MID$(NS$, FL(I), 1) THEN SOL = 0
      I = I + 1
    WEND
'      Check if unique letters correspond to unique digits
    FOR I = 1 TO NL - 1
      FOR J = I + 1 TO NL
     C1$ = MID$(NS$, UL(I), 1)
     C2$ = MID$(NS$, UL(J), 1)
     IF C1$ = C2$ THEN SOL = 0
      NEXT J
    NEXT I
'       Display Solution
    IF SOL > 0 THEN
      FOR I = 1 TO NL
     PRINT MID$(L$, UL(I), 1); " = "; MID$(NS$, UL(I), 1)
      NEXT I
      PRINT : TOT = TOT + 1: END  ' Only one needed
    END IF
  NEXT N2
NEXT N1
IF TOT = 0 THEN PRINT "NO SOLUTION POSSIBLE"
```