FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '91
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will display a phrase as a rectangle.
'
A$ = "COMPUTER CONTEST 1991"
CLS
PRINT A$: L = LEN(A$)
FOR I = 2 TO L - 1
  LOCATE I, 1: PRINT MID$(A$, I, 1)
  LOCATE I, L: PRINT MID$(A$, L - I + 1, 1)
NEXT I
FOR I = L TO 1 STEP -1
  PRINT MID$(A$, I, 1);
NEXT I


'1.2
' This program will display 2 random #s and their sum.
'
RANDOMIZE TIMER
X = INT(RND(3) * 19) - 9
Y = INT(RND(3) * 19) - 9
PRINT X; " + "; Y; " = "; X + Y


'1.3
' This program prints the total point score for a team.
'
INPUT "Enter team name: "; N$
INPUT "Enter # of 1 point programs: "; P1
INPUT "Enter # of 2 point programs: "; P2
INPUT "Enter # of 3 point programs: "; P3
TOT = P1 + P2 * 2 + P3 * 3
PRINT N$; " SCORED"; TOT; "POINTS"


'1.4
' This program displays a spreadsheet.
'
CLS
PRINT "   A B C D E F G H I J K L M N O P Q R S T"
FOR I = 1 TO 20: PRINT USING "##"; I: NEXT I


'1.5
' This program determines the number of teams competing.
'
INPUT "Enter number of students: "; X
PRINT X / 4; "TEAMS"
```

```
'1.6
' This program displays a word twice intersecting at a letter.
'
INPUT "Enter word: "; A$
INPUT "Enter letter: "; L$
X = INSTR(A$, L$)
CLS : LOCATE X, 1: PRINT A$
FOR I = 1 TO LEN(A$)
  LOCATE I, X:  PRINT MID$(A$, I, 1)
NEXT I


'1.7
' This program displays fields from an account key.
'
INPUT "Enter account key: "; A$
PRINT "ORGANIZATION "; MID$(A$, 1, 3)
PRINT "BRANCH "; MID$(A$, 4, 3)
PRINT "DEALER "; MID$(A$, 7, 4)
PRINT "CLASS "; MID$(A$, 11, 3)
PRINT "UNIT "; MID$(A$, 14, 6)


'1.8
' This program displays the # of job steps in JCL.
'
INPUT "Enter line: "; L$
WHILE L$ <> "//"
  IF L$ = "EXEC" THEN ST = ST + 1
  INPUT "Enter line: "; L$
WEND
PRINT ST; "JOB STEPS"


'1.9
' This program will replace MAN with PERSON.
'
INPUT "Enter sentence: "; S$
FOR I = 1 TO LEN(S$)
  M$ = MID$(S$, I, 3)
  IF M$ = "MAN" THEN
    PRINT "PERSON"; : I = I + 2
  ELSE
    IF M$ = "MEN" THEN
      PRINT "PERSONS"; : I = I + 2
    ELSE
      PRINT MID$(S$, I, 1);
    END IF
  END IF
NEXT I
```

```
'1.10
' This program determines the winner of two computer teams.
'
INPUT "Enter team name: "; N1$
INPUT "Enter points, time, penalties: "; P1, T1, PEN1
'
INPUT "Enter team name: "; N2$
INPUT "Enter points, time, penalties: "; P2, T2, PEN2
'
IF P1 > P2 THEN W$ = N1$ ELSE IF P2 > P1 THEN W$ = N2$
H1 = INT(T1 / 100):   M1 = T1 - H1 * 100
H2 = INT(T2 / 100):   M2 = T2 - H2 * 100
TI1 = H1 * 60 + M1 + PEN1 * 5
TI2 = H2 * 60 + M2 + PEN2 * 5
IF P1 = P2 THEN IF TI1 < TI2 THEN W$ = N1$ ELSE W$ = N2$
PRINT W$; " WINS"
```

```
'2.1
' This program displays a pyramid of consecutive numbers.
'
INPUT "Enter N: "; N
S = 1
WHILE S < N
  I = I + 1
  PRINT SPACE$(20 - I * 2);
  FOR J = 1 TO I
    PRINT MID$(STR$(100 + S), 3, 2); "  ";
    S = S + 1
  NEXT J
  PRINT
WEND


'2.2
' This program will line up numbers with decimal points.
'
FOR I = 1 TO 5
  INPUT "Enter #: "; A$(I)
NEXT I
FOR I = 1 TO 5
  X = INSTR(A$(I), ".")
  PRINT SPACE$(6 - X); A$(I)
  SUM# = SUM# + VAL(A$(I))
NEXT I
PRINT " ---------"
TOT$ = STR$(SUM# + .00001)          'Round off machine error
X = INSTR(TOT$, ".")
PRINT SPACE$(6 - X); MID$(TOT$, 1, X + 4)  'Round off error
```

```
'2.3
' This program will convert BASIC to COBOL.
'
INPUT "Enter statement: "; S$
FOR I = 1 TO LEN(S$)
  M$ = MID$(S$, I, 1):  N$ = MID$(S$, I + 1, 1)
  MN$ = MID$(S$, I, 2)
  IF M$ = "=" OR M$ = ">" OR M$ = "<" THEN
    IF N$ = "=" OR N$ = ">" OR N$ = "<" THEN
      IF MN$ = "<=" OR MN$ = "=<" THEN PRINT "IS NOT GREATER
THAN";
      IF MN$ = ">=" OR MN$ = "=>" THEN PRINT "IS NOT LESS THAN";
      IF MN$ = "<>" OR MN$ = "><" THEN PRINT "IS NOT EQUAL TO";
      I = I + 1
    ELSE
      IF M$ = ">" THEN PRINT "IS GREATER THAN";
      IF M$ = "<" THEN PRINT "IS LESS THAN";
      IF M$ = "=" THEN PRINT "IS EQUAL TO";
    END IF
  ELSE
    PRINT M$;
  END IF
NEXT I


'2.4
' This program ranks teams in a league.
'
INPUT "Enter N: "; N
FOR I = 1 TO N
  INPUT "Enter team: "; N$(I)
  INPUT "Enter wins, losses: "; W(I), L(I)
NEXT I
FOR I = 1 TO N - 1
  FOR J = I + 1 TO N
    IF W(I) <= W(J) OR (W(I) = W(J) AND N$(I) > N$(J)) THEN
      SWAP W(I), W(J): SWAP L(I), L(J): SWAP N$(I), N$(J)
    END IF
  NEXT J
NEXT I
' Display teams in order
FOR I = 1 TO N
  IF W(I) = W(I - 1) THEN PRINT R;  ELSE PRINT : PRINT I; : R = I
  PRINT N$(I); SPACE$(13 - LEN(N$(I))); W(I); ","; L(I)
NEXT I
```

```
'2.5
' This program will guess a secret number within 7 tries.
'
INC = 64: GUESS = 64
WHILE A$ <> "R"
  G = G + 1
  PRINT USING "GUESS #"; G;
  PRINT ":"; GUESS
  INPUT "Enter H, L, or R: "; A$
  INC = INC / 2
  IF A$ = "L" THEN GUESS = GUESS - INC
  IF A$ = "H" THEN GUESS = GUESS + INC
WEND



'2.6
' This program prints text in pyramid form.
'
INPUT "Enter text: "; A$:  L = LEN(A$)
I = 1
WHILE I <= L
  MD$ = MID$(A$, I, 1)
  IF MD$ <> " " THEN
    L$ = L$ + MD$
  ELSE
    IF LEN(L$) < PL + 2 THEN
      L$ = L$ + MD$
    ELSE
      PL = LEN(L$)
      PRINT SPACE$(20 - INT(PL / 2)); L$: L$ = ""
    END IF
  END IF
  I = I + 1
WEND
PRINT SPACE$(20 - INT(LEN(L$) / 2)); L$


'2.7
' This program displays a rectangle of asterisks.
'
INPUT "Enter length, width: "; L, W
CLS
COL = INT((80 - L) / 2):  ROW = INT((24 - W) / 2)
LOCATE ROW, COL
FOR I = 1 TO L:  PRINT "*"; : NEXT I
FOR I = 1 TO W - 2
  LOCATE ROW + I, COL:  PRINT "*"
  LOCATE ROW + I, COL + L - 1:  PRINT "*"
NEXT I
LOCATE ROW + W - 1, COL
FOR I = 1 TO L:  PRINT "*"; : NEXT I
```

```
'2.8
' This program displays a bar graph for lengths.
'
DIM A(12)
INPUT "Enter title: "; T$
FOR I = 0 TO 11
  PRINT "Enter # for"; 1980 + I; ":";
  INPUT A(I): IF A(I) > MAX THEN MAX = A(I)
NEXT I
INC = MAX / 20
CLS : PRINT SPACE$(3); T$; SPACE$(3);
PRINT USING "ASTERISK = ####.##"; INC
FOR I = 20 TO 1 STEP -1: PRINT USING "##"; I: NEXT I
FOR I = 1 TO 12 * 3 + 2: PRINT "-"; : NEXT I
PRINT : PRINT "  ";
FOR I = 0 TO 11: PRINT USING "###"; 80 + I; : NEXT I
FOR I = 0 TO 11
  FOR J = 1 TO INT(A(I) / INC)
    LOCATE 22 - J, I * 3 + 4: PRINT " *"
  NEXT J
NEXT I
LOCATE 23, 1
```

```
'2.9
' This program displays a store maintenance list.
'
INPUT "Enter # of entries in yesterday's file: "; F1
FOR I = 1 TO F1
  INPUT "Enter ID: "; ID1$(I)
  INPUT "Enter item: "; ITEM1$(I)
NEXT I
INPUT "Enter # of entries in today's file: "; F2
FOR I = 1 TO F2
  INPUT "Enter ID: "; ID2$(I)
  INPUT "Enter item: "; ITEM2$(I)
NEXT I
PRINT : PRINT "ADDED"
FOR I = 1 TO F2
  J = 1
  WHILE J < F1 AND ID2$(I) <> ID1$(J): J = J + 1: WEND
  IF ID2$(I) <> ID1$(J) THEN
    AN = AN + 1: PRINT ID2$(I); " "; ITEM2$(I)
  END IF
NEXT I
PRINT : PRINT "CHANGED"
FOR I = 1 TO F1
  J = 1
  WHILE J < F2 AND (ID1$(I) <> ID2$(J) OR ITEM1$(I) = ITEM2$(J))
    J = J + 1
  WEND
  IF ID1$(I) = ID2$(J) AND ITEM1$(I) <> ITEM2$(J) THEN
    CN = CN + 1: PRINT ID1$(I); " "; ITEM1$(I); " "; ITEM2$(J)
  END IF
NEXT I
PRINT : PRINT "DELETED"
FOR I = 1 TO F1
  J = 1
  WHILE J < F2 AND ID1$(I) <> ID2$(J): J = J + 1: WEND
  IF ID1$(I) <> ID2$(J) THEN
    DN = DN + 1: PRINT ID1$(I); " "; ITEM1$(I)
  END IF
NEXT I
PRINT
PRINT "TOTAL ADDED ="; AN
PRINT "TOTAL CHANGED ="; CN
PRINT "TOTAL DELETED ="; DN
```

```
'2.10
' This program displays the contents of contest diskettes.
'
INPUT "Enter year: "; Y$:  YY$ = RIGHT$(Y$, 2): Y = VAL(YY$)
DATA PRB,JDG,PG1,PG2,BAS,PAS
FOR I = 1 TO 6: READ Z$(I): NEXT I
XXX$(1) = "ONE": XXX$(2) = "TWO": XXX$(3) = "THR"
FOR I = 1 TO 4
  FOR J = 1 TO 3
    PRINT "FHS"; YY$; "-"; MID$(STR$(J), 2); "."; Z$(I)
  NEXT J
NEXT I
TOT = 12
FOR I = 5 TO 6
  FOR J = 1 TO 3
    P = 10
    IF Y = 80 AND J = 3 THEN P = 12
    IF Y = 81 THEN P = 5
    IF Y = 82 AND J = 2 THEN P = 12
    IF Y = 82 AND J = 3 THEN P = 8
    FOR K = 1 TO P
      PRINT XXX$(J); MID$(STR$(K), 2); "T"; YY$; "."; Z$(I)
      TOT = TOT + 1
      IF TOT = 20 THEN
        A$ = "": WHILE A$ = "": A$ = INKEY$: WEND: TOT = 0
      END IF
    NEXT K
  NEXT J
NEXT I
```

```
'3.1
' This program simulates a baseball game.
'
DEFINT A-W
RANDOMIZE TIMER
CLS : PRINT
PRINT SPACE$(8);
FOR I = 1 TO 9: PRINT I; : NEXT I: PRINT " SCORE"
PRINT SPACE$(8); : FOR I = 1 TO 33: PRINT "-"; : NEXT I: PRINT
PRINT "TEAM A !"; SPACE$(27); "!"
PRINT "TEAM B !"; SPACE$(27); "!"
FOR IN = 1 TO 9
  FOR T = 1 TO 2
    S = 0: B = 0: W = 0: R = 0: O = 0
    WHILE O < 3
      X = RND(3)
      IF X < .4 THEN S = S + 1: STOT = STOT + 1
      IF X >= .4 THEN B = B + 1: BTOT = BTOT + 1
      IF S = 3 THEN O = O + 1: OTOT = OTOT + 1: S = 0: W = 0
      IF B = 4 THEN W = W + 1: WTOT = WTOT + 1: B = 0: S = 0
      IF W = 4 THEN R = R + 1: R(T) = R(T) + 1: W = 3
    WEND
    LOCATE 3 + T, 6 + IN * 3: PRINT R;
  NEXT T
NEXT IN
LOCATE 4, 39: PRINT USING "##"; R(1)
LOCATE 5, 39: PRINT USING "##"; R(2)
PRINT
PRINT "TOTAL # OF STRIKES:"; STOT
PRINT "TOTAL # OF BALLS:"; BTOT
PRINT "TOTAL # OF WALKS:"; WTOT
PRINT "TOTAL # OF STRIKE OUTS:"; OTOT


'3.2
' This program displays the units digit in a power expression.
'
DEFINT A-Z
INPUT "Enter A, X: "; A(1), X(1)
INPUT "Enter B, Y: "; A(2), X(2)
INPUT "Enter C, Z: "; A(3), X(3)
FOR I = 1 TO 3
  POW = 1
  FOR J = 1 TO X(I)
    POW = POW * A(I)
    C = INT(POW / 10)
    POW = POW - C * 10
   NEXT J
   SUM = SUM + POW
NEXT I
C = INT(SUM / 10)
PRINT SUM - C * 10
```

```
'3.3
' This program displays all digits in X ^ Y.
'
DEFINT A-Z
DIM A(200)
INPUT "Enter X, Y: "; X, Y
A(1) = 1: DIG = 1
FOR I = 1 TO Y
  FOR J = 1 TO DIG
    A(J) = A(J) * X + C
    C = INT(A(J) / 10)
    A(J) = A(J) - C * 10
  NEXT J
  WHILE C > 0
    CC = INT(C / 10):   DIG = DIG + 1
    A(DIG) = C - CC * 10:  C = CC
  WEND
NEXT I
FOR I = DIG TO 1 STEP -1
  PRINT USING "#"; A(I);
NEXT I
```

```
'3.4
' This program assigns user LOGON IDs to names.
'
INPUT "Enter name: "; N$(1):  T = 1
WHILE N$(T) <> "END"
  T = T + 1
  INPUT "Enter name: "; N$(T)
WEND
' Extract parts of name for initials
T = T - 1
FOR I = 1 TO T
  FOR J = 1 TO LEN(N$(I))
    MD$ = MID$(N$(I), J, 1)
    IF MD$ <> " " THEN
      W$ = W$ + MD$
    ELSE
      IF F = 1 THEN M$(I) = W$: M = 1
      IF F = 0 THEN F$(I) = W$: F = 1
      W$ = ""
    END IF
  NEXT J
  IF M = 0 THEN M$(I) = "X"
  L$(I) = W$: W$ = "": F = 0: M = 0
  INIT$(I) = LEFT$(F$(I), 1) + LEFT$(M$(I), 1) + LEFT$(L$(I), 1)
  IN2$(I) = INIT$(I): N2$(I) = L$(I) + " " + F$(I): C(I) = I
NEXT I
' Sort Initials
FOR I = 1 TO T - 1
  FOR J = I + 1 TO T
     IF IN2$(I) > IN2$(J) THEN
       SWAP IN2$(I), IN2$(J): SWAP N2$(I), N2$(J): SWAP C(I), C(J)
     END IF
  NEXT J
NEXT I
' Sort names within same initials and assign numbers
J = 0
WHILE J < T - 1
  I = J + 1: J = I + 1
  WHILE (IN2$(I) <> IN2$(J)) AND (I < T)
    I = I + 1:  J = J + 1
  WEND
  WHILE (IN2$(I) = IN2$(J)): J = J + 1: WEND: J = J - 1
  FOR A = I TO J - 1
    FOR B = A + 1 TO J
      IF N2$(A) > N2$(B) THEN
        SWAP N2$(A), N2$(B):  SWAP C(A), C(B)
      END IF
    NEXT B
  NEXT A
' Assign numbers for middle initial
  FOR A = I TO J
    MID$(INIT$(C(A)), 2, 1) = MID$(STR$(A - I + 1), 2, 1)
  NEXT A
WEND
FOR I = 1 TO T
```

```
  PRINT N$(I); SPACE$(19 - LEN(N$(I))); "SD"; INIT$(I); "1"
NEXT I


'3.5
' This program displays the digits 0 - 9 in enlarged form.
'   1    The data contains the
'  2 3   line segment #s (on the left)
'   4    that need to be displayed to
'  5 6   produce the corresponding
'   7    digits: 0,1,2,3,4,5,6,7,8,9
DATA 123567,36,13457,13467,2346,12467,124567,136,1234567,12346
FOR N = 0 TO 9
  CLS : READ A$
  FOR J = 1 TO LEN(A$)
    X = VAL(MID$(A$, J, 1))
    SELECT CASE X
      CASE 1: LOCATE 1, 1: PRINT STRING$(11, "*")
      CASE 2: FOR I = 1 TO 8: LOCATE I, 1: PRINT "*": NEXT I
      CASE 3: FOR I = 1 TO 8: LOCATE I, 11: PRINT "*": NEXT I
      CASE 4: LOCATE 8, 1: PRINT STRING$(11, "*")
      CASE 5: FOR I = 1 TO 8: LOCATE I + 7, 1: PRINT "*": NEXT I
      CASE 6: FOR I = 1 TO 8: LOCATE I + 7, 11: PRINT "*": NEXT I
      CASE 7: LOCATE 15, 1: PRINT STRING$(11, "*")
    END SELECT
  NEXT J
  SLEEP (1)
NEXT N


'3.6
' This program will evaluate an expression with ().
'
INPUT "Enter expression: "; A$
FOR I = 1 TO LEN(A$)
  M$ = MID$(A$, I, 1)
  IF M$ = "(" THEN P = P + 1: P1(P) = S + 1
  IF M$ = "+" OR M$ = "-" THEN S = S + 1: SY$(S) = M$
  IF M$ >= "0" AND M$ <= "9" THEN N = N + 1: NUM(N) = VAL(M$)
  IF M$ = ")" THEN
    FOR J = P1(P) TO S
      IF SY$(J) = "-" THEN NUM(J + 1) = NUM(J) - NUM(J + 1)
      IF SY$(J) = "+" THEN NUM(J + 1) = NUM(J) + NUM(J + 1)
    NEXT J
    N = P1(P): NUM(N) = NUM(S + 1)
    S = P1(P) - 1: P = P - 1
  END IF
NEXT I
FOR I = 1 TO S
  IF SY$(I) = "-" THEN NUM(I + 1) = NUM(I) - NUM(I + 1)
  IF SY$(I) = "+" THEN NUM(I + 1) = NUM(I) + NUM(I + 1)
NEXT I
PRINT NUM(N)
```

```
'3.7
' This program displays the two pay days for a given month.
'
DIM MON(12), MNAME$(12)
DATA MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,SUNDAY
DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY,AUGUST
DATA SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
DATA 31,28,31,30,31,30,31,31,30,31,30,31
FOR I = 1 TO 7: READ DNAME$(I): NEXT I
FOR I = 1 TO 12: READ MNAME$(I): NEXT I
FOR I = 1 TO 12: READ MON(I): NEXT I
H = 1
INPUT "Enter holiday MM, DD: "; MHOL(H), DHOL(H)
WHILE MHOL(H) > 0
  H = H + 1
  INPUT "Enter holdiay MM, DD: "; MHOL(H), DHOL(H)
WEND
H = H - 1
PRINT : INPUT "Enter month #: "; MNUM: PRINT
WHILE MNUM > 0
  DAYS(1) = 0
  FOR I = 1 TO MNUM - 1
    DAYS(1) = DAYS(1) + MON(I)
  NEXT I
  DAY(1) = 15:  DAY(2) = MON(MNUM)
  DAYS(2) = DAYS(1) + DAY(2)
  DAYS(1) = DAYS(1) + DAY(1)
  FOR T = 1 TO 2
    HOL = 1
'   Decrement days counters if holiday or weekend
    WHILE HOL = 1 OR WKEND = 1
      HOL = 0:  WKEND = 0
      FOR I = 1 TO H
        IF MHOL(I) = MNUM AND DAY(T) = DHOL(I) THEN
          DAY(T) = DAY(T) - 1: DAYS(T) = DAYS(T) - 1: HOL = 1
        END IF
      NEXT I
      X = DAYS(T) MOD 7
      IF X = 5 OR X = 6 THEN       '5 = Saturday or 6 = Sunday
        DAY(T) = DAY(T) - 1: DAYS(T) = DAYS(T) - 1: WKEND = 1
      END IF
    WEND
    PRINT DNAME$(X + 1); " "; MNAME$(MNUM); DAY(T)
  NEXT T
  PRINT : INPUT "Enter month #: "; MNUM: PRINT
WEND
```

```
'3.8
' This program will display 3 x 3 magic squares.
'
INPUT "Enter digit: "; DIG
INPUT "Enter row, col: "; ROW, COL
DATA 6,7,2
DATA 1,5,9
DATA 8,3,4
FOR I = 1 TO 3: FOR J = 1 TO 3: READ A(I, J): NEXT J, I
ROT = 1
WHILE (A(ROW, COL) <> DIG) AND (ROT < 4)
'  Rotate outer numbers clockwise, at most 3 times
   X = A(1, 1): A(1, 1) = A(3, 1): A(3, 1) = A(3, 3)
   A(3, 3) = A(1, 3): A(1, 3) = X
   X = A(1, 2): A(1, 2) = A(2, 1): A(2, 1) = A(3, 2)
   A(3, 2) = A(2, 3): A(2, 3) = X
   ROT = ROT + 1
WEND
IF A(ROW, COL) <> DIG THEN PRINT "NO SOLUTION": END
FOR P = 1 TO 2
  FOR I = 1 TO 3
    FOR J = 1 TO 3
      PRINT A(I, J);
    NEXT J: PRINT
  NEXT I: PRINT
  IF P = 1 THEN
    IF (ROW = 1 AND COL = 2) OR (ROW = 3 AND COL = 2) THEN
'   Swap with respect to 2nd column
      SWAP A(1, 1), A(1, 3): SWAP A(2, 1), A(2, 3)
      SWAP A(3, 1), A(3, 3)
    END IF
    IF (ROW = 1 AND COL = 1) OR (ROW = 3 AND COL = 3) THEN
'   Swap with respect to main diagonal
      SWAP A(1, 2), A(2, 1): SWAP A(1, 3), A(3, 1)
      SWAP A(3, 2), A(2, 3)
    END IF
    IF (ROW = 1 AND COL = 3) OR (ROW = 3 AND COL = 1) THEN
'   Swap with respect to minor diagonal
      SWAP A(2, 1), A(3, 2): SWAP A(1, 1), A(3, 3)
      SWAP A(1, 2), A(2, 3)
    END IF
    IF (ROW = 2 AND COL = 1) OR (ROW = 2 AND COL = 3) THEN
'   Swap with respect to 2nd row
      SWAP A(1, 1), A(3, 1): SWAP A(1, 2), A(3, 2)
      SWAP A(1, 3), A(3, 3)
    END IF
  END IF
NEXT P
```

```
'3.9
' This program will display a pie graph.
'
DIM A(21, 21)
INPUT "Enter 3 percentages: "; P(1), P(2), P(3)
A$(1) = "A": A$(2) = "D": A$(3) = "N"
CLS : PI = 3.14159
' Draw circle
FOR I = -PI / 2 TO 3 / 2 * PI STEP .1
  X = COS(I) * 10:  Y = SIN(I) * 10
  LOCATE 11 + Y, 11 + X: PRINT "*": A(11 + Y, 11 + X) = 1
NEXT I
' Draw 3 line segments from center
FOR S = 0 TO 2
  SUM = SUM + P(S)
  I = -PI / 2 + 2 * PI * SUM / 100
  FOR R = 0 TO 10
    X = COS(I) * R: Y = SIN(I) * R
    LOCATE 11 + Y, 11 + X: PRINT "*": A(11 + Y, 11 + X) = 1
  NEXT R
NEXT S
A$ = INPUT$(1):  SUM = 0
' Fill regions with letters
FOR S = 1 TO 3
  LSUM = SUM:  SUM = SUM + P(S)
  FOR L = LSUM TO SUM
    I = -PI / 2 + 2 * PI * L / 100
    FOR R = 1 TO 9
      X = COS(I) * R: Y = SIN(I) * R
      IF A(11 + Y, 11 + X) = 0 THEN
        LOCATE 11 + Y, 11 + X: PRINT A$(S)
      END IF
    NEXT R
  NEXT L
NEXT S
```

```
'3.10
' This program will convert large numbers in base 2,4,8,16.
'
DEFINT A-Z
DIM A(255)
INPUT "Enter numeral: "; NUM$
INPUT "Enter base M: "; M
INPUT "Enter base N: "; N
L = LEN(NUM$)
DIGM = INT(LOG(M) / LOG(2) + .001)
DIGN = INT(LOG(N) / LOG(2) + .001)
PAD = DIGN - (DIGM * L MOD DIGN): IF PAD = DIGN THEN PAD = 0
FOR I = 1 TO PAD: A(I) = 0: NEXT I
' Convert from base M to base 2
FOR I = 1 TO L
  D$ = MID$(NUM$, I, 1)
  NUM = INSTR("0123456789ABCDEF", D$) - 1
  FOR J = DIGM - 1 TO 0 STEP -1
    X = INT(NUM / 2 ^ J)
    IND = I * DIGM - J + PAD
    A(IND) = X
    NUM = NUM - X * 2 ^ J
  NEXT J
NEXT I
' Convert from base 2 to base N
LIND = DIGM * L + PAD: ZERO = 1
FOR I = 0 TO (LIND / DIGN) - 1
  SUM = 0
  FOR J = 1 TO DIGN
    IND = I * DIGN + J
    SUM = SUM + A(IND) * 2 ^ (DIGN - J)
  NEXT J
  IF ZERO = 0 OR SUM > 0 THEN
    ZERO = 0
    PRINT MID$("0123456789ABCDEF", SUM + 1, 1);
  END IF
NEXT I
```