FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '87
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will print out the sign of a given number.
'
INPUT "Enter number:"; N
IF N > 0 THEN PRINT "POSITIVE"
IF N < 0 THEN PRINT "NEGATIVE"
IF N = 0 THEN PRINT "ZERO"
```

```
'1.2
' This program will sum the numbers n, n+1, ... n+20.
'
INPUT "Enter n:"; N
FOR I = 0 TO 20
  SUM = SUM + N + I
NEXT I
PRINT "SUM ="; SUM
```

```
'1.3
' This program will print PROBLEM THREE diagonally.
'
CLS
A$ = "PROBLEM THREE"
L = LEN(A$)
ROW = (24 - L) \ 2: COL = (80 - L) \ 2
FOR I = 1 TO L
  LOCATE ROW + I, COL + I: PRINT MID$(A$, I, 1)
NEXT I
```

```
'1.4
' This program displays the numbers on the sides of a die.
'
INPUT "Enter number on top:"; T
INPUT "Enter number on front:"; F
INPUT "Enter number on right:"; R
PRINT "TOP="; T
PRINT "FRONT="; F
PRINT "RIGHT="; R
PRINT "BOTTOM="; 7 - T
PRINT "BACK="; 7 - F
PRINT "LEFT="; 7 - R
```

```
'1.5
' This program will fill the screen with random characters.
'
CLS
FOR I = 1 TO 24
  FOR J = 1 TO 80
    X = INT(RND(3) * 96) + 33
    PRINT CHR$(X);
  NEXT J
NEXT I
WHILE A$ = "": A$ = INKEY$: WEND
CLS


'1.6
' This program will display a rectangular array of periods.
'
INPUT "Enter coordinates:"; UR, UC, LR, LC
CLS
FOR I = UR TO LR
  FOR J = UC TO LC
    LOCATE I, J: PRINT ".";
  NEXT J
NEXT I


'1.7
' This program will generate 10 random numbers given a seed.
'
INPUT "Enter seed:"; SEED
FOR I = 1 TO 10
  RAND = (SEED * 421 + 1)
  RAND = RAND - INT(RAND / 100) * 100
  SEED = RAND
  PRINT RAND
NEXT I


'1.8
' This program will determine the mass of a fish tank.
'
INPUT "Enter K, L, W, H:"; K, L, W, H
MASS = L * 12 * 2.54 * W * 12 * 2.54 * H * 12 * 2.54
MASS = MASS / 1000 + K
PRINT USING "#####.## KILOGRAMS"; MASS;
```

```
'1.9
' This program will display 21 rows of letters.
'
CLS
FOR I = 1 TO 21
  IF I MOD 2 = 1 THEN
    PRINT STRING$(31, 64 + I)
  ELSE
    FOR J = 1 TO 10
      PRINT CHR$(64 + I); SPACE$(2);
    NEXT J
    PRINT CHR$(64 + I)
  END IF
NEXT I


'1.10
' This program will display the time needed to read a book.
'
DATA THE HISTORY OF THE COMPUTER,400
DATA THE RED DOG RUNS,200
DATA EATING APPLE PIE,150
DATA THE ART OF WINNING,250
INPUT "Enter book title:"; B$
INPUT "Enter rate (minutes/page):"; SP
I = 0
WHILE (I < 4) AND (A$ <> B$)
  READ A$, PA
  I = I + 1
WEND
M = PA * SP
H = INT(M / 60): M = M - H * 60
PRINT H; "HOURS "; M; "MINUTES"
```

```
'2.1
' This program will rotate a string N times to the left.
'
INPUT "Enter string: "; S$
INPUT "Enter N:"; N
L = LEN(S$)
N = N MOD L
PRINT RIGHT$(S$, L - N); LEFT$(S$, N)


'2.2
' This program will determine the number of diskettes bought.
'
FOR V = 1 TO 98
  FOR M = 1 TO 99 - V STEP 5
    W = 100 - V - M
    IF W >= 0 AND V * 225 + M * 297 + W * 120 = 23607 THEN
      PRINT V; "VERS "; M; "MAXS "; W; "WABS": END
    END IF
  NEXT M
NEXT V


'2.3
' This program will display a subset of random numbers.
'
DIM A(15)
RANDOMIZE TIMER
INPUT "Enter list item:"; ITEM
WHILE ITEM <> -1
  A(J) = ITEM
  INPUT "Enter list item:"; ITEM
  J = J + 1
WEND
WHILE A$ <> CHR$(27)
   FOR I = 0 TO 4
     SWAP A(I), A(INT(RND * (J - I) + I))
     PRINT A(I)
   NEXT I
   PRINT "PRESS ANY KEY": A$ = ""
   WHILE A$ = "": A$ = INKEY$: WEND
WEND
```

```
'2.4
' This program will display all partitioned sum of number.
'
INPUT "Enter a number less than 20:"; N
FOR I = N TO 1 STEP -1
  IF N MOD I = 0 THEN
    X = INT(N / I)
    PRINT SPACE$(N - X);
    I$ = MID$(STR$(I), 2)
    PRINT I$;
    IF I < N THEN
      FOR J = 1 TO X - 1
     PRINT "+"; I$;
      NEXT J
    END IF
    PRINT
  END IF
NEXT I


'2.5
' This program will calculate the fractional value.
'
INPUT "Enter word: "; A$
FOR I = 1 TO 3
  A(I) = ASC(MID$(A$, I, 1)) - 64
NEXT I
N = A(1) * A(2) + A(2) * A(3) + A(1) * A(3)
D = A(1) * A(2) * A(3)
FOR I = D TO 1 STEP -1
  IF N MOD I = 0 AND D MOD I = 0 THEN
    PRINT LTRIM$(STR$(N / I)); "/"; LTRIM$(STR$(D / I)): END
  END IF
NEXT I
```

```
'2.6
' This program will find a subset of integers.
'
I = 1
INPUT "Enter set item:"; ITEM(I)
WHILE ITEM(I) > 0
  I = I + 1
  INPUT "Enter set item:"; ITEM(I)
WEND
LASTI = I - 1
INPUT "Enter N:"; N
INPUT "Enter S:"; S
' Sort list
FOR I = 1 TO LASTI - 1
  FOR J = I + 1 TO LASTI
    IF ITEM(I) > ITEM(J) THEN SWAP ITEM(I), ITEM(J)
  NEXT J
NEXT I
SUM = 0
FOR I = 1 TO N: SUM = SUM + ITEM(I): NEXT I
IF SUM > S THEN PRINT "NO": END
PRINT " YES"
FOR I = 1 TO N: PRINT ITEM(I); : NEXT I
```

```
'2.7
' This program will determine if patterns are legal/illegal.
'
DATA 1,4,3,4,4,5
DATA 5,2,5,2,5,5
FOR I = 0 TO 5: READ A(I): NEXT I
FOR I = 0 TO 5: READ B(I): NEXT I
INPUT "Enter pattern:"; P$
STATE = 0
'
' Run the state machine
'
LP = LEN(P$)
FOR I = 1 TO LP       'Check whole string even if error found
  C$ = MID$(P$, I, 1)
  IF C$ <> "A" AND C$ <> "B" THEN STATE = 5  'illegal pattern
  IF C$ = "A" THEN STATE = A(STATE) ELSE STATE = B(STATE)
NEXT I
IF STATE = 4 THEN PRINT "LEGAL PATTERN": END
PRINT "ILLEGAL PATTERN"
```

```
'2.8
' This program will find integers having F factors.
'
INPUT "Enter M, N, F:"; M, N, F
FOR I = M TO N
  S = 0: X = INT(SQR(I) + .000001)
  FOR J = 1 TO X
    IF I MOD J = 0 THEN S = S + 2
  NEXT J
  IF X * X = I THEN S = S - 1
  IF S = F THEN PRINT I
NEXT I
```

```
'2.9
' This program will alphabetize 5 words according to rules.
'
DIM A$(12), B$(12), C$(12)
FOR I = 1 TO 5
  INPUT "Enter word: "; A$(I): L = LEN(A$(I))
  FOR J = 1 TO L
    B$(J) = MID$(A$(I), J, 1)
  NEXT J
' Alphabetize letters within word to make word2 (C$)
  FOR J = 1 TO L - 1
    FOR K = J + 1 TO L
      IF B$(J) > B$(K) THEN SWAP B$(J), B$(K)
    NEXT K
    C$(I) = C$(I) + B$(J)
  NEXT J
  C$(I) = C$(I) + B$(L)
NEXT I
' Alphabetize words according to word2 (C$)
FOR I = 1 TO 4
  FOR J = I + 1 TO 5
    IF C$(I) > C$(J) THEN SWAP C$(I), C$(J): SWAP A$(I), A$(J)
  NEXT J
NEXT I
FOR I = 1 TO 5: PRINT A$(I): NEXT I
```

```
'2.10
' This program will produce a super-duper input routine
' with 4 types of input.
'
INPUT "Enter ROW, COL:"; ROW, COL
INPUT "Enter MAX:"; MAX
INPUT "Enter TYPE:"; TYP
CLS : CH$ = " ": INITCOL = COL
DO UNTIL CH$ = CHR$(13)
  LOCATE ROW, COL: CH$ = ""
  WHILE CH$ = "": CH$ = INKEY$: WEND
'
  IF CH$ = CHR$(8) THEN   ' Backspace pressed
    IF LEN(ENTRY$) > 0 THEN
      ENTRY$ = LEFT$(ENTRY$, LEN(ENTRY$) - 1)
      COL = COL - 1: LOCATE ROW, COL: PRINT " ";
    END IF
  ELSE
    VALIDCH = LEN(ENTRY$) < MAX
    IF VALIDCH THEN
     SELECT CASE TYP
      CASE 1
       IF CH$ <> " " AND (CH$ < "A" OR CH$ > "Z") THEN VALIDCH = 0
      CASE 2
       IF CH$ <> "." AND (CH$ < "0" OR CH$ > "9") THEN VALIDCH = 0
      CASE 3
       IF COL - INITCOL = 2 OR COL - INITCOL = 5 THEN
         IF CH$ <> "-" THEN VALIDCH = 0
       ELSE
         IF CH$ < "0" OR CH$ > "9" THEN VALIDCH = 0
       END IF
     END SELECT
    END IF
    IF VALIDCH THEN
      PRINT CH$;
      ENTRY$ = ENTRY$ + CH$
      COL = COL + 1
    END IF
  END IF
LOOP
LOCATE ROW + 2, INITCOL: PRINT ENTRY$
```

```
'3.1
' This program will determine if 2 words are closely spelled.
'
INPUT "Enter word 1: "; W1$
INPUT "Enter word 2: "; W2$
L1 = LEN(W1$): L2 = LEN(W2$)
IF ABS(L1 - L2) > 1 THEN PRINT "NOT CLOSE": END
' Find first position where words differ
IF L1 < L2 THEN MIN = L1 ELSE MIN = L2
J = 1
WHILE (J <= MIN) AND MID$(W1$, J, 1) = MID$(W2$, J, 1)
  J = J + 1
WEND
IF J > MIN THEN PRINT "CLOSE": END  'Equal or differ by ins/del
IF L1 = L2 THEN
' Check for transposition or one symbol change
  IF J <> L1 THEN
    IF MID$(W1$, J + 1, 1) = MID$(W2$, J, 1) THEN
     IF MID$(W2$, J + 1, 1) = MID$(W1$, J, 1) THEN
       J = J + 1   'Skip over possible transposition
     END IF
    END IF
  END IF
  IF MID$(W1$, J + 1) = MID$(W2$, J + 1) THEN PRINT "CLOSE": END
  PRINT "NOT CLOSE": END
ELSE
' Check for insertion or deletion
  IF L1 > L2 THEN
    IF MID$(W2$, J) = MID$(W1$, J + 1) THEN PRINT "CLOSE": END
    PRINT "NOT CLOSE"
  ELSE
    IF MID$(W1$, J) = MID$(W2$, J + 1) THEN PRINT "CLOSE": END
    PRINT "NOT CLOSE"
  END IF
END IF
```

```
'3.2
' This program will evaluate an NxN determinant for N=2,3,4.
'
INPUT "Enter dimension N:"; N
FOR I = 1 TO N
  FOR J = 1 TO N
    PRINT USING "Enter row #"; I;
    PRINT USING ", col #:"; J; : INPUT A(I, J)
  NEXT J
NEXT I
' -- 2x2
IF N = 2 THEN
  PRINT A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1)
ELSE
' -- 3x3
  IF N = 3 THEN
    K = 4: GOSUB Det3x3
    PRINT T
  ELSE
' -- 4x4
    FOR K = 1 TO 4
      A = A(4, K) * (-1) ^ K
'
Det3x3:
      FOR I = 1 TO 3
        FOR J = 1 TO 4
          IF J <> K THEN
            S = S + 1: B(I, S) = A(I, J)
            B(I, S + 3) = A(I, J)
          END IF
        NEXT J: S = 0
      NEXT I
      FOR I = 1 TO 3
        T = T + B(1, I) * B(2, I + 1) * B(3, I + 2)
        T = T - B(1, I + 2) * B(2, I + 1) * B(3, I)
      NEXT I
      IF N = 3 THEN RETURN
'
      B = B + T * A: T = 0
    NEXT K: PRINT B
  END IF
END IF
```

```
'3.3
' This program will display the number of word occurrences.
'
DIM WORD$(50), WORDTOT(50)
INPUT "Enter text: "; LINES$
START = 1:  NUMOFWORDS = 0
WHILE START <= LEN(LINES$)
  ENDOFWORD = 0:  NEXTWORD$ = ""
  WHILE (START <= LEN(LINES$)) AND (NOT ENDOFWORD)
    CH$ = MID$(LINES$, START, 1)
    IF (CH$ < "A" OR CH$ > "Z") AND (CH$ <> "'") THEN
      ENDOFWORD = -1
    ELSE
      NEXTWORD$ = NEXTWORD$ + CH$
    END IF
      START = START + 1
  WEND
  IF NEXTWORD$ > "" THEN NEWWORD = -1 ELSE NEWWORD = 0
  WORDIND = 0
  WHILE (WORDIND < NUMOFWORDS) AND NEWWORD
    WORDIND = WORDIND + 1
    IF NEXTWORD$ = WORD$(WORDIND) THEN NEWWORD = 0
  WEND
  IF NOT NEWWORD THEN
    WORDTOT(WORDIND) = WORDTOT(WORDIND) + 1
  ELSE
'   Add new word to list
    NUMOFWORDS = NUMOFWORDS + 1
    WORD$(NUMOFWORDS) = NEXTWORD$
    WORDTOT(NUMOFWORDS) = 1
  END IF
WEND
FOR I = 1 TO NUMOFWORDS
  PRINT WORDTOT(I); WORD$(I)
NEXT I
```

```
'3.4
' This program will encrypt a string such that when this
' code is entered, the string will be reproduced.
'
DIM ASCI(30)
INPUT "Enter text: "; ST$
NUMOFCH = 0: I = 1
WHILE (I <= LEN(ST$))
  CH$ = MID$(ST$, I, 1):  NUMOFCH = NUMOFCH + 1
  IF CH$ = "\" THEN
      I = I + 1: NEXTCH$ = MID$(ST$, I, 1)
      IF NEXTCH$ = "\" THEN
        ASCI(NUMOFCH) = ASC(NEXTCH$)
      ELSE
        ASCST$ = MID$(ST$, I, 3)
        ASCI(NUMOFCH) = VAL(ASCST$)
        I = I + 2
      END IF
  ELSE
    ASCI(NUMOFCH) = ASC(CH$)   'Regular character
  END IF
  I = I + 1
WEND
'  Encrypt code
FOR I = 1 TO NUMOFCH
  CODENUM = 255 - ASCI(I)
  IF (CODENUM >= 32) AND (CODENUM <= 92) THEN
    PRINT CHR$(CODENUM);
    IF CODENUM = ASC("\") THEN PRINT "\";
  ELSE
    PRINT "\";
    PRINT MID$(STR$(1000 + CODENUM), 3, 3);
  END IF
NEXT I
```

```
'3.5
' This program will unscrabmle the numbers 5132, 4735, and
' 8014153 so that the first times the second equals the
' third wit a missing digit.
'
DIM D(24), E(24)
DATA 5,1,3,2
DATA 4,7,3,5
DATA 8,0,1,4,1,5,3
FOR I = 1 TO 4: READ A(I): NEXT I
FOR I = 1 TO 4: READ B(I): NEXT I
FOR I = 1 TO 7: READ B$(I): NEXT I
FOR A = 1 TO 4
  FOR B = 1 TO 4
    FOR C = 1 TO 4: D = 10 - A - B - C
      D = 4 + 3 + 2 + 1 - A - B - C
      IF A <> B AND B <> C AND A <> C THEN
     S = S + 1
     D(S) = A(A) * 1000 + A(B) * 100 + A(C) * 10 + A(D)
     E(S) = B(A) * 1000 + B(B) * 100 + B(C) * 10 + B(D)
      END IF
    NEXT C
  NEXT B
NEXT A
FOR I = 1 TO 24
  FOR J = 1 TO 24
    X# = D(I) * E(J)
    A$ = LTRIM$(STR$(X#))
    IF LEN(A$) = 8 THEN
      FOR K = 1 TO 8
     A$(K) = MID$(A$, K, 1)
      NEXT K
      B = 1: MATCH = -1
      WHILE (B <= 7) AND MATCH
     MATCH = 0: A = 1
     WHILE (A <= 8) AND NOT MATCH
       IF B$(B) = A$(A) THEN A$(A) = "": MATCH = -1
       A = A + 1
     WEND
     B = B + 1
      WEND
      IF MATCH THEN PRINT D(I); E(J); " "; A$
    END IF
  NEXT J
NEXT I
```

```
'3.6
' This program will display the front colors on the Rubik's
' Pocket Cube after a move of T or F is performed.
'
DIM A$(24)
DATA W,Y,O,G,R,B
FOR I = 1 TO 6: READ A$
  FOR J = 1 TO 4
    S = S + 1: A$(S) = A$
  NEXT J
NEXT I
INPUT "Enter T, F, or Q: "; A$
DO UNTIL A$ = "Q"
  IF A$ = "T" THEN
'   Rotate Top
    X$ = A$(1): A$(1) = A$(3): A$(3) = A$(4)
    A$(4) = A$(2): A$(2) = X$
    X$ = A$(5): A$(5) = A$(9): A$(9) = A$(13)
    A$(13) = A$(17): A$(17) = X$
    X$ = A$(6): A$(6) = A$(10): A$(10) = A$(14)
    A$(14) = A$(18): A$(18) = X$
  ELSE
'   Rotate Front
    X$ = A$(5): A$(5) = A$(7): A$(7) = A$(8)
    A$(8) = A$(6): A$(6) = X$
    X$ = A$(3): A$(3) = A$(20): A$(20) = A$(22)
    A$(22) = A$(9): A$(9) = X$
    X$ = A$(4): A$(4) = A$(18): A$(18) = A$(21)
    A$(21) = A$(11): A$(11) = X$
  END IF
' Display front side
  PRINT A$(5); "  "; A$(6)
  PRINT A$(7); "  "; A$(8)
  INPUT "Enter T, F, or Q: "; A$
LOOP
```

```
'3.7
' This program will simulate a drill of adding Roman Numerals.
'
CLS
INPUT "Enter name: "; NME$
INPUT "Enter date: "; DAYTE$
'
DATA M,1000,D,500,C,100,L,50,X,10,V,5,I,1
FOR I = 1 TO 7: READ B$(I), B(I): NEXT I
'
DO UNTIL A$ = "3"
  CLS
  PRINT "1. INSTRUCTION PAGE"
  PRINT "2. PRACTICE 3 PROBLEMS"
  PRINT "3. QUIT"
  A$ = INPUT$(1)
  SELECT CASE A$
    CASE "1"
      CLS
      PRINT "YOU WILL BE GIVEN 3 PROBLEMS TO"
      PRINT "WORK. A PROBLEM WILL CONSIST OF"
      PRINT "ADDING TWO RANDOMLY GENERATED"
      PRINT "ROMAN NUMERALS LESS THAN 20."
      PRINT "YOU WILL TYPE YOUR ANSWER IN"
      PRINT "ROMAN NUMERALS AND PRESS 'RETURN.'"
      PRINT "(PRESS ANY KEY TO RETURN TO MENU.)"
      AN$ = INPUT$(1)
    '
    ' Practice 3 problems
    '
    CASE "2"
      RIGHT = 0: WRONG = 0
      FOR PROB = 1 TO 3
        CLS
        RANDOMIZE TIMER
        X(1) = INT(RND * 19) + 1:  X(2) = INT(RND * 19) + 1
        X(3) = X(1) + X(2):  HELP = X(3)
        FOR K = 1 TO 3: X$(K) = "": NEXT K
        FOR K = 1 TO 3
          FOR I = 1 TO 7
            X = X(K) / B(I)
            IF (ABS(X - 9 / 5) > .01) OR (I MOD 2 = 1) THEN
              X = INT(X)
              SELECT CASE X
                CASE 9
                  X$(K) = X$(K) + B$(I) + B$(I - 2)
                CASE 4
                  X$(K) = X$(K) + B$(I) + B$(I - 1)
                CASE IS > 0
                  FOR J = 1 TO X: X$(K) = X$(K) + B$(I): NEXT J
              END SELECT
              X(K) = X(K) - B(I) * X
            END IF
          NEXT I
        NEXT K
```

```
      '
      ' Display problem
      '
      LOCATE 10, 15: PRINT X$(1): X = LEN(X$(1))
      Y = LEN(X$(2)): COL = 15 + (X - Y) - 2
      LOCATE 11, COL: PRINT "+ "; X$(2)
      LOCATE 12, COL: PRINT STRING$(2 + Y, "-"): MISS = -1
      WHILE MISS <> 0
        LOCATE 13, COL: INPUT N$
        '
        ' Evaluate
        '
        IF N$ = X$(3) THEN
          RIGHT = RIGHT + 1: MISS = 0
        ELSE
          IF MISS > 0 THEN
            MISS = 0: BEEP: WRONG = WRONG + 1: WR$(WRONG) = N$
            RI$(WRONG) = X$(3): RI(WRONG) = HELP
          ELSE
            MISS = 1: BEEP: LOCATE 16, COL: PRINT HELP
            LOCATE 13, COL: PRINT SPACE$(15)
          END IF
        END IF
      WEND
    NEXT PROB
    '
    ' Progress Report
    '
    CLS : PRINT SPACE$(11); "PROGRESS REPORT"
    PRINT "DATE: "; DAYTE$
    PRINT "NAME: "; NME$
    PRINT "NUMBER CORRECT:"; RIGHT
    PRINT "NUMBER OF EXERCISES: 3"
    PRINT "PERCENT CORRECT:"; INT(RIGHT / 3 * 100 + .5); "%"
    PRINT
    IF WRONG > 0 THEN
      LOCATE 15, 1: PRINT "WRONG ANSWER   CORRECT ANSWER   ARABIC"
      FOR I = 1 TO WRONG
        LOCATE 16 + I, 1: PRINT WR$(I)
        LOCATE 16 + I, 16: PRINT RI$(I)
        LOCATE 16 + I, 32: PRINT RI(I)
      NEXT I
    END IF
    LOCATE 23, 1: PRINT "PRESS ANY KEY TO RETURN TO MENU.";
    AN$ = INPUT$(1)
  END SELECT
LOOP
```

```
'3.8
' This program will determine the area shared w/2 rectangles.
'
DIM AB(20, 20), XY(20, 20)
FOR I = 1 TO 4
  INPUT "Enter X,Y: "; X(I), Y(I)
  X(I) = ABS(X(I)): Y(I) = ABS(Y(I))
NEXT I
FOR I = 1 TO 4
  INPUT "Enter A,B: "; A(I), B(I)
  A(I) = ABS(A(I)): B(I) = ABS(B(I))
NEXT I
'
' Store a 1 in each occupied square of AB
'
FOR I = A(1) TO A(2)
  FOR J = B(4) TO B(1)
    AB(I, J) = 1
  NEXT J
NEXT I
'
' Determine area in common (Heighth-1 x Width-1)
'
FOR I = X(1) TO X(2)
  FOR J = Y(4) TO Y(1)
    IF AB(I, J) = 1 THEN WDTH = WDTH + 1    'Both interior
  NEXT J
  IF WDTH > 0 THEN HEIGHT = HEIGHT + 1: WDTH2 = WDTH: WDTH = 0
NEXT I
PRINT (HEIGHT - 1) * (WDTH2 - 1)
```

```
'3.9
' This program will divide 2 big numbers with at most 30 digits.
'
DIM A(30), B(30), C(30), D(30)
INPUT "Enter first number: "; A$: LENA = LEN(A$)
INPUT "Enter second number:"; B$: LENB = LEN(B$)
L = LENB
'
'  Store digits in arrays
'
FOR I = LENB TO 1 STEP -1
 B(LENB - I + 1) = VAL(MID$(B$, I, 1))
NEXT I
FOR I = LENB TO 1 STEP -1
  A(LENB - I + 1) = VAL(MID$(A$, I, 1))
NEXT I: K = LENB
'
' Shift digits of A until portion of A is greater than B
'
ShiftDigits:
   IF L <> LENB THEN
NextShift:
     K = K + 1: IF LENA < K THEN GOTO DisplayRemainder
     FOR I = L TO 1 STEP -1
       A(I + 1) = A(I)
     NEXT I
     A(1) = VAL(MID$(A$, K, 1))
     L = L + 1
     IF L < LENB THEN PRINT "0"; : GOTO NextShift
   END IF
   IF L <= LENB THEN
     FOR I = LENB TO 1 STEP -1
       IF A(I) > B(I) THEN GOTO DivideAbyB
       IF A(I) <> B(I) THEN GOTO NextShift
     NEXT I
     ' All A(I) = B(I) at this point
   END IF
'
' Divide A by B by subtracting B * J from A
'
DivideAbyB:
  SUBDONE = 0
  DO
    J = J + 1
    FOR I = 1 TO LENB
      C(I) = B(I) * J + C
      C = INT(C(I) / 10)
      C(I) = C(I) - C * 10
    NEXT I: C(I) = C: C = 0
    FOR I = 1 TO L
      D(I) = A(I) - C(I) - D
      D = -(D(I) < 0): IF D THEN D(I) = D(I) + 10
    NEXT I
    IF L - LENB = 0 OR D(L) = 0 THEN
      I = LENB + 1
```

```
      DO
        I = I - 1
        IF D(I) < B(I) THEN SUBDONE = -1
      LOOP UNTIL I = 1 OR D(I) > B(I) OR SUBDONE
    END IF
  LOOP UNTIL SUBDONE
  '
  ' Display J as # of subtractions done
  '
  PRINT USING "#"; J; : L = 0: J = 0
  FOR I = LENB TO 1 STEP -1
    IF D(I) > 0 OR T > 0 THEN
      T = 1: L = L + 1: A(I) = D(I)
    END IF
  NEXT I: T = 0: GOTO ShiftDigits
'
'  Display remainder
'
DisplayRemainder:
  PRINT " Remainder ";
  FOR I = L TO 1 STEP -1
    PRINT USING "#"; A(I);
  NEXT I
  IF L = 0 THEN PRINT "0"
```

```
'3.10
' This program will generate random mazes with 3 x 5 paths.
'
CLS : RANDOMIZE TIMER: L = 8: W = 5
NUMOFLINES = (L - 1) * (W - 1)  '# of lines to draw
LI = INT(32 / L)
WI = INT(15 / W)
LN = L: WN = W
DIM A(LN + 1, WN + 1)   'Points forbidden to start from
DIM PINT(33, 33)        'Existing points
'
' Draw perimeter
'
FOR I = 1 TO 33: PRINT "*"; : PINT(I - 1, 0) = 1: NEXT I
FOR I = 1 TO 14
  LOCATE I + 1, 1: PRINT "*": PINT(0, I) = 1
  LOCATE I + 1, 33: PRINT "*": PINT(L, I) = 1
NEXT I
FOR I = 1 TO 33: PRINT "*"; : PINT(I - 1, W) = 1: NEXT I
'
A(0, 0) = 1: A(LN, 0) = 1: A(LN, WN) = 1: A(0, WN) = 1
DO
  DO
  ' Get point that exists but is not forbidden
    X = INT(RND * 2 * LN) - INT(LN / 2)
    Y = INT(RND * 2 * WN) - INT(WN / 2)
    IF X < 0 THEN X = 0
    IF X > LN THEN X = LN
    IF Y < 0 THEN Y = 0
    IF Y > WN THEN Y = WN
  LOOP UNTIL (PINT(X, Y) = 1 AND A(X, Y) = 0)
  DO
    D = INT(RND * 4)                    'Random direction
    SEGMENTDRAWN = 0: NUMOFTRIES = 0
    DO
      NUMOFTRIES = NUMOFTRIES + 1
      D = D + 1: IF D > 4 THEN D = D - 4
      SELECT CASE D
        '
        ' Up
        '
        CASE 1
          IF Y > 0 THEN
            IF PINT(X, Y - 1) = 0 THEN
              FOR J = 0 TO WI - 1
                LOCATE Y * WI - J, X * LI + 1: PRINT "*"
              NEXT J
              A = X: B = Y - 1: SEGMENTDRAWN = -1
            END IF
          END IF
        '
        ' Right
        '
        CASE 2
          IF X < LN THEN
```

```
            IF PINT(X + 1, Y) = 0 THEN
              FOR J = 0 TO LI - 1
                LOCATE Y * WI + 1, X * LI + 2 + J: PRINT "*"
              NEXT J
              A = X + 1: B = Y: SEGMENTDRAWN = -1
            END IF
          END IF
        '
        ' Down
        '
        CASE 3
          IF Y < WN THEN
            IF PINT(X, Y + 1) = 0 THEN
              FOR J = 0 TO WI - 1
                LOCATE Y * WI + 2 + J, X * LI + 1: PRINT "*"
              NEXT J
              A = X: B = Y + 1: SEGMENTDRAWN = -1
            END IF
          END IF
        '
        ' Left
        '
        CASE 4
          IF X > 0 THEN
            IF PINT(X - 1, Y) = 0 THEN
              FOR J = 0 TO LI - 1
                LOCATE Y * WI + 1, X * LI - J: PRINT "*"
              NEXT J
              A = X - 1: B = Y: SEGMENTDRAWN = -1
            END IF
          END IF
      END SELECT
    LOOP UNTIL SEGMENTDRAWN OR (NUMOFTRIES = 4)
      '
      IF SEGMENTDRAWN THEN
        PINT(A, B) = 1: LINESDRAWN = LINESDRAWN + 1
         X = A: Y = B
      ELSE
        A(X, Y) = 1
      END IF
    LOOP UNTIL (LINESDRAWN = NUMOFLINES) OR NOT SEGMENTDRAWN
  LOOP UNTIL (LINESDRAWN = NUMOFLINES)
'
' Open doors
'
X = INT(RND * WN) + 1: Y = INT(RND * WN) + 1
FOR J = 0 TO WI - 2
  LOCATE X * WI - J, 1: PRINT " "
  LOCATE Y * WI - J, 33: PRINT " "
NEXT J
LOCATE 23
```