FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '86
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will print "THIS IS THE EASIEST PROGRAM!".
'
CLS
A$ = "THIS IS THE EASIEST PROGRAM!"
LOCATE 12, (80 - LEN(A$)) / 2: PRINT A$
```

```
'1.2
' This program will display the sum, difference, and product.
'
INPUT "Enter two numbers: "; A, B
PRINT "SUM = "; A + B
PRINT "DIFFERENCE = "; A - B
PRINT "PRODUCT = "; A * B
```

```
'1.3
' This program will sum 1 + (1/2)^2 + (1/3)^3 + (1/4)^4 + ...
' until the difference between it and the next term is within E.
'
INPUT "Enter test value E: "; E
I = 1
SUM = 1:  LSUM = 0
WHILE (SUM - LSUM) >= E
  I = I + 1
  TRM = 1 / I:  PROD = 1
  FOR J = 1 TO I: PROD = PROD * TRM: NEXT J
  LSUM = SUM
  SUM = SUM + PROD
WEND
PRINT USING "#.######"; LSUM
```

```
'1.4
' This program will print a check given name and amount.
'
CLS
INPUT "Enter first name: "; F$
INPUT "Enter middle name: "; M$
INPUT "Enter last name: "; L$
I$ = LEFT$(M$, 1)
INPUT "Enter amount: "; AMOUNT$
'  Display border
LOCATE 6, 1
PRINT STRING$(39, "*")
FOR I = 1 TO 9
  LOCATE 6 + I, 1: PRINT "*"
  LOCATE 6 + I, 39: PRINT "*"
NEXT I
PRINT STRING$(39, "*")
'
LOCATE 8, 3: PRINT "BEN'S TOWING SERVICE"
LOCATE 9, 3: PRINT "4563 WRECKER AVENUE"
LOCATE 10, 3: PRINT "WAVERLY, ARKANSAS 45632"
LOCATE 12, 4: PRINT "PAY TO THE ORDER OF ";
PRINT F$; " "; I$; ". "; L$
LOCATE 14, 4: PRINT "THE SUM OF $"; AMOUNT$
LOCATE 22, 1
```

```
'1.5
' This program will determine which prisoners may be released.
'
DIM CELL(100)
FOR I = 1 TO 100: CELL(I) = 1: NEXT I    'Cells initially open
FOR I = 2 TO 100
  J = 1
  WHILE J <= 100
    CELL(J) = 1 - CELL(J):  J = J + I
  WEND
NEXT I
FOR I = 1 TO 100
  IF CELL(I) = 1 THEN PRINT "CELL"; I
NEXT I
```

```
'1.6
' This program will determine how much money accumulates.
' Double precision variables (#) are needed.
'
INPUT "Enter monthly investment: "; MONTH#
INPUT "Enter end of year deposit: "; DEP#
INPUT "Enter annual rate of interest: "; RATE#
PRINT
RATE# = RATE# / (12 * 100) 'Rate per month in yr in percent
FOR YEAR = 1 TO 20
  FOR J = 1 TO 12
    SUM# = SUM# + MONTH#
    SUM# = SUM# + RATE# * SUM#
   NEXT J
   SUM# = SUM# + DEP#
NEXT YEAR
SUM# = INT(SUM# * 100 + .5) / 100
PRINT "AMOUNT AT END OF YEAR 20 IS $"; LTRIM$(STR$(SUM#))
```

```
'1.7
' This program will drop g in words ending with ing or ings.
'
INPUT "Enter sentence: "; S$
S$ = S$ + " "
L = LEN(S$): W$ = ""
FOR I = 1 TO L
  CH$ = MID$(S$, I, 1)
  IF CH$ <> " " THEN
    W$ = W$ + CH$
  ELSE
    LENW = LEN(W$)
    IF LENW >= 4 THEN
      EN1$ = MID$(W$, LENW - 2, 3)
      EN2$ = MID$(W$, LENW - 3, 4)
      IF EN1$ = "ING" THEN W$ = MID$(W$, 1, LENW - 1)
      IF EN2$ = "INGS" THEN W$ = MID$(W$, 1, LENW - 2) + "S"
    END IF
    PRINT W$; " ";
    W$ = ""
  END IF
NEXT I
```

```
'1.8
' This program simulates the population growth of rabbits.
'
INPUT "Enter initial population: "; INIT
INPUT "Enter point of over population: "; OP
PRINT
POP = INIT
DIEING = (POP >= OP)
FOR MONTH = 1 TO 23
  IF DIEING THEN
    IF POP < 2 / 3 * INIT THEN
      POP = POP + POP * .2: DIEING = 0
    ELSE
      POP = POP - POP * .15
    END IF
  ELSE
    IF POP >= OP THEN
      DIEING = -1:  INIT = INT(POP)
      POP = POP - POP * .15
    ELSE
      POP = POP + POP * .2
    END IF
  END IF
  PRINT "POPULATION FOR MONTH"; MONTH; "IS"; INT(POP + .5)
NEXT MONTH
```

```
'1.9
' This program doubles every e that appears as a single e.
'
INPUT "Enter sentence: "; SENT$
FOR I = 1 TO LEN(SENT$)
  CH$ = MID$(SENT$, I, 1)
  NCH$ = MID$(SENT$, I + 1, 1)
  IF CH$ = "E" AND LCH$ <> "E" AND NCH$ <> "E" THEN PRINT "E";
  PRINT CH$;
  LCH$ = CH$
NEXT I
IF NCH$ = "E" AND LCH$ <> "E" THEN PRINT "E";
PRINT NCH$
```

```
'1.10
' This program will display common elements of two lists.
'
DIM A(12), B(12), C(12)
FOR I = 1 TO 12
  PRINT "Enter"; I; "of 12: "; : INPUT A(I)
NEXT I
FOR I = 1 TO 11
  PRINT "Enter"; I; "of 11: "; : INPUT B(I)
NEXT I
'
FOR I = 1 TO 12
  FOR J = 1 TO 11
    IF A(I) = B(J) THEN C(I) = 1
  NEXT J
NEXT I
FOR I = 1 TO 12
  FOR J = I + 1 TO 12
    IF A(I) = A(J) AND C(J) > 0 THEN C(J) = C(J) + 1
  NEXT J
NEXT I
FOR I = 1 TO 12
  IF C(I) = 1 THEN PRINT A(I); "  ";
NEXT I
```

```
'2.1
' This program will right justify sentence within 65 columns.
'
COL = 65
INPUT "Enter sentence: "; SENT$
SENT$ = SENT$ + " ":  L = LEN(SENT$)
I = 1: WN = 1: WORD$(WN) = "":  TOTCH = 0
WHILE I <= L
  CH$ = MID$(SENT$, I, 1)
  IF CH$ <> " " THEN
    WORD$(WN) = WORD$(WN) + CH$
  ELSE
    IF WORD$(WN) <> "" THEN
      TOTCH = TOTCH + LEN(WORD$(WN))
      WN = WN + 1:  WORD$(WN) = ""
    END IF
  END IF
  I = I + 1
WEND
WN = WN - 1
'
SPAVE = INT((COL - TOTCH) / (WN - 1))
EXTRA = (COL - TOTCH) - (SPAVE * (WN - 1))
FOR I = 1 TO WN
  IF I <= EXTRA THEN EX = 1 ELSE EX = 0
  PRINT WORD$(I); SPACE$(SPAVE + EX);
NEXT I
```

```
'2.2
' This program will produce a repeating patern of XXX ---.
'
INPUT "Enter total number of X's and -'s: "; TOTALXD
INPUT "Enter number of X's: "; NUMX
INPUT "Enter number of rows: "; ROWS
X1$ = "": X2$ = "": D1$ = "": D2$ = ""
FOR I = 1 TO NUMX
  X1$ = X1$ + "X"
  D2$ = D2$ + "-"
NEXT I
FOR I = 1 TO TOTALXD - NUMX
  X2$ = X2$ + "X"
  D1$ = D1$ + "-"
NEXT I
FOR ROW = 1 TO ROWS
  IF ROW - INT(ROW / 2) * 2 = 1 THEN
    FOR I = 1 TO 4: PRINT X1$; D1$; : NEXT I
  ELSE
    FOR I = 1 TO 4: PRINT D2$; X2$; : NEXT I
  END IF
  PRINT
NEXT ROW
```

```
'2.3
' This program will code or decode a message.
'
ST1$ = "ZXCVBNMASDFGHJKLQWERTYUIOP "
ST2$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ "
WHILE OP < 3
  PRINT
  PRINT "1) ENCODE"
  PRINT "2) DECODE"
  PRINT "3) END"
  INPUT "Choose: "; OP
  IF OP = 3 THEN END
  INPUT "Enter message: "; MESSAGE$
  FOR I = 1 TO LEN(MESSAGE$)
    CH$ = MID$(MESSAGE$, I, 1)
    IF CH$ <> " " THEN
      IF OP = 1 THEN
        CH$ = MID$(ST1$, ASC(CH$) - 64, 1)
      ELSE
        J = INSTR(ST1$, CH$)
        CH$ = MID$(ST2$, J, 1)
      END IF
    END IF
    PRINT CH$;
  NEXT I
  PRINT
WEND
```

```
'2.4
' This program finds the unique mode of a set of 15 numbers.
'
DIM A(15), C(15)
FOR I = 1 TO 15
  PRINT "Enter number"; I; ": "; : INPUT A(I)
NEXT I
MAX = 1
FOR I = 1 TO 14
  C(I) = 1
  FOR J = I + 1 TO 15
    IF A(I) = A(J) THEN
      C(I) = C(I) + 1
      IF C(I) > MAX THEN MAX = C(I)
    END IF
  NEXT J
NEXT I
MODEXIST = 0
FOR I = 1 TO 14
  IF C(I) = MAX THEN
    IF MODEXIST THEN PRINT "NO UNIQUE MODE": END
    MODE = A(I): MODEXIST = -1
  END IF
NEXT I
IF MODEXIST THEN PRINT "MODE IS"; MODE: END
PRINT "NO UNIQUE MODE"
```

```
'2.5
' This program simulates transactions to savings a account.
'
RATE = .07
INPUT "Enter original balance: "; BALANCE
WHILE OP < 4
  PRINT
  PRINT "1. MAKE A DEPOSIT"
  PRINT "2. MAKE A WITHDRAWAL"
  PRINT "3. CREDIT INTEREST"
  PRINT "4. END"
  INPUT "Enter option: "; OP
  PRINT
  IF OP = 1 THEN
    INPUT "Enter amount to deposit: "; DEP
    PRINT USING "BALANCE BEFORE TRANSACTION $####.##"; BALANCE
    BALANCE = BALANCE + DEP
    PRINT "MAKE A DEPOSIT"
  ELSEIF OP = 2 THEN
    INPUT "Enter amount to withdraw: "; WIT
    PRINT USING "BALANCE BEFORE TRANSACTION $####.##"; BALANCE
    BALANCE = BALANCE - WIT
    PRINT "MAKE A WITHDRAWAL"
  ELSEIF OP = 3 THEN
    PRINT USING "BALANCE BEFORE TRANSACTION $####.##"; BALANCE
    CREDIT = BALANCE * RATE / 12
    CREDIT = INT(CREDIT * 100 + .5) / 100
    PRINT USING "CREDIT INTEREST OF $##.##"; CREDIT
    BALANCE = BALANCE + CREDIT
  END IF
  IF OP < 4 THEN PRINT "NEW ";  ELSE PRINT "FINAL ";
  PRINT USING "BALANCE $####.##"; BALANCE
WEND
```

```
'2.6
' This program will sum two positive big numbers.
'
DIM A(39), B(39), C(39)
INPUT "ENTER FIRST NUMBER:  "; ST1$
INPUT "ENTER SECOND NUMBER: "; ST2$
L1 = LEN(ST1$):  L2 = LEN(ST2$)
FOR I = 1 TO L1
  CH$ = MID$(ST1$, L1 - I + 1, 1)
  A(I) = VAL(CH$)
NEXT I
FOR I = 1 TO L2
  CH$ = MID$(ST2$, L2 - I + 1, 1)
  B(I) = VAL(CH$)
NEXT I
'
IF L1 > L2 THEN MAXL = L1 ELSE MAXL = L2
FOR I = 1 TO MAXL
  C(I) = A(I) + B(I) + CARRY
  IF C(I) > 9 THEN C(I) = C(I) - 10: CARRY = 1 ELSE CARRY = 0
NEXT I
IF CARRY = 1 THEN MAXL = MAXL + 1: C(MAXL) = 1
PRINT "SUM IS ";
FOR I = MAXL TO 1 STEP -1
  PRINT USING "#"; C(I);
NEXT I
```

```
'2.7
' This program will perform conversions.
'
DATA "INCHES","FEET","MILES","OUNCES","POUNDS","GALLONS"
FOR I = 1 TO 6:  READ DEC$(I): NEXT I
DATA 2.54, 0.3048, 1.6093, 28.35, 0.4536, 3.7854
FOR I = 1 TO 6: READ CON(I): NEXT I
DATA "CENTIMETERS", "METERS", "KILOMETERS", "GRAMS"
DATA "KILOGRAMS", "LITERS"
FOR I = 1 TO 6: READ MET$(I): NEXT I
'
WHILE OP <> 13
  PRINT
  FOR I = 1 TO 6
    PRINT I;
    IF I - INT(I / 2) * 2 = 1 THEN
      ST$ = MET$(INT((I + 1) / 2)) + " TO "
      ST$ = ST$ + DEC$(INT((I + 1) / 2))
      PRINT ST$; SPACE$(23 - LEN(ST$));
      PRINT USING "## "; I + 6;
      ST$ = MET$(INT((I + 7) / 2)) + " TO "
      ST$ = ST$ + DEC$(INT((I + 7) / 2))
    ELSE
      ST$ = DEC$(INT(I / 2)) + " TO "
      ST$ = ST$ + MET$(INT(I / 2))
      PRINT ST$; SPACE$(23 - LEN(ST$));
      PRINT USING "## "; I + 6;
      ST$ = DEC$(INT((I + 6) / 2)) + " TO "
      ST$ = ST$ + MET$(INT((I + 6) / 2))
    END IF
    PRINT ST$
  NEXT I
  PRINT SPACE$(26); "13 END"
  INPUT "Enter option: "; OP
  IF OP < 13 THEN
    IF OP - INT(OP / 2) * 2 = 1 THEN
      PRINT "Enter number of "; MET$(INT((OP + 1) / 2));
      INPUT ": "; X
      Y = X / CON(INT((OP + 1) / 2))
      PRINT USING "THIS IS EQUIVALENT TO ###.### "; Y;
      PRINT DEC$(INT((OP + 1) / 2))
    ELSE
      PRINT "Enter number of "; DEC$(INT(OP / 2));
      INPUT ": "; X
      Y = X * CON(INT(OP / 2))
      PRINT USING "THIS IS EQUIVALENT TO ###.### "; Y;
      PRINT MET$(INT(OP / 2))
    END IF
  END IF
WEND
```

```
'2.8
' This program will generate a mortgage amortization.
' Double precision variables are needed.
'
INPUT "Enter principal: "; PRINC#
INPUT "Enter % rate of interest: "; RATE#
INPUT "Enter term in years: "; YEARS
INPUT "Enter # of month in year for first payment: "; MONTH
RATE# = RATE# / (12 * 100): AMOUNT# = 1
FOR I = 1 TO YEARS * 12: AMOUNT# = AMOUNT# * (1 + RATE#): NEXT I
PAYMENT# = (RATE# * AMOUNT#) / (AMOUNT# - 1) * PRINC#
C = MONTH - 1:  OLDP# = PRINC#
RATE# = RATE# * 12
PRINT "INTEREST         PRINCIPAL"
'
FOR I = 1 TO YEARS * 12
  MI# = OLDP# * RATE# / 12
  MP# = PAYMENT# - MI#
  OLDP# = OLDP# - MP#
  PRINT USING "$###.##"; MI#; : PRINT SPACE$(10);
  PRINT USING "$#####.##"; OLDP#
  C = C + 1:  YI# = YI# + MI#
  IF C - INT(C / 12) * 12 = 0 THEN
    PRINT
    PRINT USING "YEAR'S INTEREST  $#####.##"; YI#
    TI# = TI# + YI#:  YI# = 0
    PRINT
    A$ = INPUT$(1)
  END IF
NEXT I
IF MONTH <> 1 THEN
  PRINT
  PRINT USING "YEAR'S INTEREST  $#####.##"; YI#
  TI# = TI# + YI#
END IF
PRINT USING "TOTAL INTEREST   $#####.##"; TI#
PRINT USING "MONTHLY PAYMENT  $#####.##"; PAYMENT#
```

```
'2.9
' This program calculates the value of sine(x) by a series.
' Double precision variables are needed.
'
INPUT "Enter N degrees: "; N
PI# = 3.1415926535#
IF N > 180 THEN X# = PI# * ((360 - N) / 180)
IF N <= 180 THEN X# = PI# * (N / 180)
POWER = -1
FOR I = 1 TO 6
  POWER = POWER + 2:  FACT = 1
  FOR J = 1 TO POWER: FACT = FACT * J: NEXT J
  TRM# = 1
  FOR J = 1 TO POWER: TRM# = TRM# * X#: NEXT J
  TRM# = TRM# / FACT
  IF I - INT(I / 2) * 2 = 1 THEN
    SUM# = SUM# + TRM#
  ELSE
    SUM# = SUM# - TRM#
  END IF
NEXT I
IF N > 180 THEN SUM# = -1 * SUM#:  X# = PI# * (N / 180)
PRINT "PARTIAL SUM ="; :       IF SUM# < 0 THEN PRINT " ";
PRINT USING "##.#######"; SUM#
PRINT "ACTUAL SINE ="; :       IF SIN(X#) < 0 THEN PRINT " ";
PRINT USING "##.#######"; SIN(X#)
```

```
'2.10
' This program will convert a Roman Numeral to Arabic form.
'
DATA M,1000, D,500, C,100, L,50, X,10, V,5, I,1
FOR I = 1 TO 7: READ RN$(I), RV(I): NEXT I
INPUT "Enter Roman Numeral: "; ROMNUM$
L = LEN(ROMNUM$):  I = 1:  ARABIC = 0
WHILE I < L
  FOR J = 1 TO 7
  IF MID$(ROMNUM$, I, 1) = RN$(J) THEN IND1 = J
  IF MID$(ROMNUM$, I + 1, 1) = RN$(J) THEN IND2 = J
NEXT J
  IF IND1 <= IND2 THEN
    ARABIC = ARABIC + RV(IND1)
  ELSE
    ARABIC = ARABIC + RV(IND2) - RV(IND1): I = I + 1
  END IF
  I = I + 1
WEND
IF I = L THEN
  FOR J = 1 TO 7
    IF MID$(ROMNUM$, I, 1) = RN$(J) THEN IND1 = J
  NEXT J
  ARABIC = ARABIC + RV(IND1)
END IF
PRINT "ARABIC ="; ARABIC
```

```
'3.1
' This program produces montly calendars for the year 1986.
'
DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
DATA AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
DATA 31,28,31,30,31,30,31,31,30,31,30,31
DATA S,M,T,W,T,F,S
DIM MO$(12), DAYS(12)
FOR I = 1 TO 12: READ MO$(I): NEXT I
FOR I = 1 TO 12: READ DAYS(I): NEXT I
FOR I = 1 TO 7:  READ D$(I): NEXT I
CLS : PRINT SPACE$(12); "1986": PRINT
FOR M = 1 TO 12
  IF M > 1 THEN CLS
  PRINT SPACE$(13 - INT(LEN(MO$(M)) / 2)); MO$(M): PRINT
  FOR I = 1 TO 7: PRINT " "; D$(I); "  "; : NEXT I
  PRINT
'
  IF M = 1 THEN COL = 4
  IF COL > 1 THEN PRINT SPACE$((COL - 1) * 4);
  FOR DAY = 1 TO DAYS(M)
    PRINT USING "##"; DAY; : PRINT "  ";
    IF COL < 7 THEN COL = COL + 1 ELSE COL = 1: PRINT
  NEXT DAY
  A$ = "":  WHILE A$ = "": A$ = INKEY$: WEND
NEXT M
```

```
'3.2
' This program finds the root of a 5th degree polynomial
' of the form Ax^5 + Bx^4 + Cx^3 + Dx^2 + Ex + F = 0.
'
INPUT "Enter coefficients A,B,C,D,E,F: "; A, B, C, D, E, F
DEF FNY (Y) = C * Y ^ 3 + D * Y * Y + E * Y + F
DEF FNP (X) = A * X ^ 5 + B * X ^ 4 + FNY(X)
' This algorithm finds 1 and only 1 root (closest to x=0)
X1 = -1:  X2 = 1
' Find sign change between X1 and X2
WHILE FNP(X1) * FNP(X2) > 0
  X1 = X1 - 1:  X2 = X2 + 1
WEND
' Use binary search to find root
WHILE X2 - X1 > .000005
  X = (X1 + X2) / 2
  IF FNP(X) * FNP(X1) > 0 THEN X1 = X ELSE X2 = X
WEND
PRINT "ROOT = ";
IF X < 0 THEN PRINT "-"; : X = -X
PRINT USING "#.#####"; X
```

```
'3.3
' This program changes a number from one base to another.
'
D$ = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
INPUT "Enter base A: "; A
INPUT "Enter base B: "; B
INPUT "Enter original number: "; NUMST$
PRINT : PRINT NUMST$; " BASE"; A; "EQUALS ";
FOR I = 1 TO LEN(NUMST$)
   POW = INT(A ^ (LEN(NUMST$) - I) + .01)
   N = N + (INSTR(D$, MID$(NUMST$, I, 1)) - 1) * POW
NEXT I
POW = 1
WHILE POW <= N
  EX = EX + 1: POW = POW * B
WEND
EX = EX - 1
' Convert Num to Base B from Base 10
FOR I = EX TO 0 STEP -1
  POW = POW / B
  X = INT(N / POW + .01)
  PRINT MID$(D$, X + 1, 1);
  N = N - X * POW
NEXT I
PRINT " BASE"; B
```

```
'3.4
' This progam will update customers account by SSN's.
'
DATA 234567890,"JOHN SMITH  "
DATA "1234 ANYWHERE LANE, EXIST, KANSAS 66754   ",345.78
DATA 564783219,"GAIL HUSTON "
DATA "543 SOUTH THIRD, BIG TOWN, TEXAS 88642    ",2365.89
DATA 873421765,"TIM JONES   "
DATA "2387 PALM PLACE, NOME, ALASKA 77643       ",6754.76
DATA 543876543,"JILL RUPERTS"
DATA "4536 123RD STREET, TINY TOWN, MAINE 76765 ",45.18
DATA 345212342,"AL BROWN    "
DATA "PO BOX 234, TINSEL TOWN, CALIFORNIA 77654 ",3456.09
DATA 565656565,"KERMIT TEU  "
DATA "1234 LOST LANE, WIMPLE, WISCONSIN 66543   ",78.36
FOR I = 1 TO 6: READ SS$(I), N$(I), A$(I), B(I): NEXT I
INPUT "Enter SSN: "; SSN$
WHILE SSN$ <> "000000000"
  I = 1
  WHILE (SS$(I) <> SSN$) AND (I < 6): I = I + 1: WEND
  INPUT "Enter C for Charge or P for Payment: "; CH$
  INPUT "Enter amount of transaction: "; TRANS
  IF CH$ = "C" THEN B(I) = B(I) - TRANS
  IF CH$ = "P" THEN B(I) = B(I) + TRANS
  PRINT : PRINT USING "NEW BALANCE IS $####.##"; B(I)
  PRINT : INPUT "Enter SSN: "; SSN$
WEND
FOR I = 1 TO 5
  FOR J = I + 1 TO 6
    IF B(I) < B(J) THEN
      SWAP SS$(I), SS$(J)
      SWAP N$(I), N$(J)
      SWAP A(I), A(J)
      SWAP B(I), B(J)
    END IF
  NEXT J
NEXT I
PRINT
PRINT "SSN        NAME          ADDRESS"; SPACE$(13);
PRINT "BALANCE": PRINT
FOR I = 1 TO 6
  PR$ = SS$(I) + "  " + N$(I) + "  "
  L = LEN(PR$) - 1
  P1 = INSTR(A$(I), ",")
  P2 = INSTR(P1 + 1, A$(I), ",")
  PRINT PR$; LEFT$(A$(I), P1 - 1); SPACE$(21 - P1);
  PRINT USING "$####.##"; B(I)
  PRINT SPACE$(L); MID$(A$(I), P1 + 1, P2 - P1 - 1)
  PRINT SPACE$(L); MID$(A$(I), P2 + 1)
NEXT I
```

```
'3.5
' This program will print the product of 2 large decimals.
'
DIM A(30), B(30), PROD(50)
INPUT "Enter first number: "; ASTR$
INPUT "Enter second number: "; BSTR$
ADEC = INSTR(ASTR$, "."):  BDEC = INSTR(BSTR$, ".")
ASTR$ = LEFT$(ASTR$, ADEC - 1) + RIGHT$(ASTR$, LEN(ASTR$) - ADEC)
BSTR$ = LEFT$(BSTR$, BDEC - 1) + RIGHT$(BSTR$, LEN(BSTR$) - BDEC)
LENA = LEN(ASTR$):  LENB = LEN(BSTR$)
RDIGITS = LENA - ADEC + LENB - BDEC + 2
FOR I = LENA TO 1 STEP -1
   A(LENA - I + 1) = VAL(MID$(ASTR$, I, 1))
NEXT I
FOR I = LENB TO 1 STEP -1
  B(LENB - I + 1) = VAL(MID$(BSTR$, I, 1))
NEXT I
FOR I = 1 TO LENB
  CARRY = 0
  FOR J = 1 TO LENA
    S = I + J - 1
    PROD(S) = PROD(S) + B(I) * A(J) + CARRY
    CARRY = INT(PROD(S) / 10)
    PROD(S) = PROD(S) - CARRY * 10
  NEXT J
  IF CARRY > 0 THEN PROD(S + 1) = CARRY
NEXT I
PRINT "PRODUCT = ";
IF CARRY > 0 THEN S = S + 1
IF S <= RDIGITS THEN PRINT "0";
FOR I = S TO 1 STEP -1
  IF I = RDIGITS THEN PRINT ".";
  PRINT USING "#"; PROD(I);
NEXT I
```

```
'3.6
' This program will determine if a # can become palindrome.
'
DIM B(50), REV(50)
INPUT "Enter number: "; NUMST$
L = LEN(NUMST$)
FOR I = 1 TO L
  B(L - I + 1) = VAL(MID$(NUMST$, I, 1))
NEXT I
TRY = 0: PAL = 0
WHILE (TRY <= 23) AND (NOT PAL)
  PAL = -1
  FOR I = 1 TO INT(L / 2)
    IF B(I) <> B(L - I + 1) THEN PAL = 0
  NEXT I
' Add reverse of number to itself
  IF NOT PAL THEN
    FOR I = 1 TO L: REV(I) = B(L - I + 1): NEXT I
    CARRY = 0
    FOR I = 1 TO L
      B(I) = B(I) + REV(I) + CARRY
      CARRY = INT(B(I) / 10)
      B(I) = B(I) - CARRY * 10
    NEXT I
    IF CARRY = 1 THEN L = L + 1: B(L) = 1
    TRY = TRY + 1
  END IF
WEND
IF NOT PAL THEN PRINT "CANNOT GENERATE A PALINDROME": END
FOR I = L TO 1 STEP -1: PRINT USING "#"; B(I); : NEXT I
PRINT " IS A PALINDROME"
```

```
'3.7
' 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 USING "Co#"; COL; : PRINT ": ";
    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 ", "; LTRIM$(STR$(INT(C(ROW, N + 1) + .1)));
NEXT ROW
PRINT ")"
```

```
'3.8
' This program prints Kth, 2*Kth, and 3*Kth permutations.
'
INPUT "Enter word: "; A$:  INPUT "Enter K: "; KK:  L = LEN(A$)
FOR I = 1 TO L: A$(I) = MID$(A$, I, 1): NEXT I
' Alphabetize letters
FOR I = 1 TO L - 1
  FOR J = I + 1 TO L
    IF A$(I) > A$(J) THEN X$ = A$(I): A$(I) = A$(J): A$(J) = X$
  NEXT J
NEXT I
' Produce factorials F(I) = (I-1)!
FOR I = 1 TO L
  F = 1
  FOR J = 1 TO I - 1: F = F * J: NEXT J
  F(I) = F
NEXT I
FOR T = 1 TO 3
  K = KK * T - 1
'     Generate Kth permutation
  FOR I = L TO 1 STEP -1
    X = INT(K / F(I))
    FOR J = 1 TO L
      IF A(J) = 0 THEN
        S = S + 1: IF S > X THEN A(J) = 1: PRINT A$(J); : J = L
      END IF
    NEXT J
    S = 0: K = K - F(I) * X
  NEXT I
  FOR I = 1 TO L: A(I) = 0: NEXT I
  PRINT "  ";
NEXT T
```

```
'3.9
' This program will solve cryptarithm puzzle ABB - CB = DEF.
' F = 0 since B-B = 0.  A=D+1 or A=D since CB is 2 digits,
' but A<>D.  D>B, otherwise D=A.  Since B<C, B<9, => E=10+B-C
'
FOR B = 1 TO 8
  FOR C = B + 1 TO 9
    FOR D = 1 TO 8
      F = 0: A = D + 1:  E = 10 + B - C
      IF A = B OR A = C OR A = D OR A = E OR A = F THEN PASS = 1
      IF B = C OR B = D OR B = E OR B = F OR C = D THEN PASS = 1
      IF C = E OR C = F OR D = E OR D = F THEN PASS = 1
      IF PASS = 0 THEN
        TOT = TOT + 1
        PRINT A * 100 + B * 10 + B; "-"; C * 10 + B; "=";
        PRINT D * 100 + E * 10 + F; " NUMBER"; TOT
      ELSE
        PASS = 0
      END IF
    NEXT D
  NEXT C
NEXT B
PRINT : PRINT " TOTAL NUMBER OF SOLUTIONS ="; TOT
```

```
'3.10
' This program will find all 2-digit integers equal to the sum
' of integers in which each digit 0-9 is used exactly once.
'
FOR I = 0 TO 8
'   Place digit I infront of 0 and sum the rest of the digits
  SUM = I * 10 + 0
  FOR J = 0 TO 9
    IF (I <> J) AND (J <> 0) THEN
      TRM = J: SUM = SUM + J
    END IF
  NEXT J
  IF SUM <= 99 THEN
'    Display sum followed by example sum process
    PRINT SUM; "=";
    PRINT I * 10 + 0;
    FOR J = 0 TO 9
      IF (I <> J) AND (J <> 0) THEN
        TRM = J: PRINT "+"; J;
      END IF
    NEXT J
    PRINT
  END IF
NEXT I
```