FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '80

1.1 Write A Program (W.A.P.) to print on the screen the terms of the Fibonacci sequence. (1,1,2,3,5,8,...). (The user will ask for the number of terms.) The sequence has the property that each number (beyond the first two) is the sum of the two previous numbers.

1.2 W.A.P. that will flash a symbol on and off approximately every 10 seconds.

1.3 W.A.P. that when given the length and width of a rectangle, determines the perimeter.

1.4 W.A.P. to convert Celsius to Fahrenheit. C = 5/9 * (F-32)

1.5 W.A.P. to test the input to see if it is a numeral or a non-numeral.

1.6 W.A.P. to determine the gas mileage of a car given the distance traveled and the number of gallons used.

1.7 W.A.P. to test if a number is divisible by five.

1.8 W.A.P. to determine one leg of a right triangle, given the other leg and the hypotenuse.

1.9 W.A.P. that will move a dot or blob across the screen.

2.1 Write A Program (W.A.P.) that will read a sequence of numbers that the operator inputs and prints the largest number in the sequence.

2.2 W.A.P. to tell whether the numbers input are the sides of a triangle, square, or rectangle.

2.3 W.A.P. that will determine the sum of the integers from 1,000 to 2,000 inclusive (i.e. 1,000 + 1,001 + ... 2,000).

2.4 W.A.P. that will reverse a 3 digit number that is given to the computer (i.e. 324 becomes 423)

2.5 W.A.P. that will draw a rectangle somewhere on the screen.

2.6 W.A.P that when given 3 numbers, will print them in increasing order.

2.7 W.A.P. to determine the mean (average) of a set of numerals.

2.8 W.A.P. to determine if a number is even or odd.

2.9 W.A.P. to determine if a number less than 200 is prime.

2.10 W.A.P. to determine the value of the change in the users's pocket. Have the computer ask for the number of each type of coin and then calculate the value.

2.11 W.A.P. that will determine the number of times the letter "e" appears in a sentence that is given to the computer.

3.1 W.A.P. to randomly generate basic multiplication facts and tell whether the operator is right or wrong.

3.2 W.A.P. to generate the digits 0-9 in random order without repetition (i.e. 1,2,0,7,4,3,6,5,9,8).

3.3 W.A.P. that will round a number to the nearest ten (i.e. 12 becomes 10; 1065 becomes 1070).

3.4 W.A.P. to change a number from base ten to base four.

3.5 W.A.P. to change a base three number to base ten.

3.6 W.A.P. to change a repeating or terminating decimal to a fraction in simplified form (i.e. .25 = 1/4).

3.7 W.A.P. that will show how to represent an amount of money less than a dollar using a minimum number of coins (i.e. $.78 = 1 $.50 piece, 1 quarter, 3 pennies).

3.8 W.A.P. where the computer selects a number from 1 to 10 and the operator attempts to guess the number. The operator is informed as to whether the guess is too low, too high, or right on.

3.9 W.A.P. that will find the values of a,b,c and d so that
a^b x c^d = abcd where abcd is a four digit number.

3.10 W.A.P. that will tell what day of the week a given date falls on in 1980. (Hint- Jan. 1 is a Tuesday).

3.11 W.A.P. that will simulate an "etch-a-sketch" (i.e. when appropriate keys are pressed, the dot traces a line segment in the appropriate direction).

3.12 W.A.P. that will determine if a word is a palindrome (i.e. ANNA is spelled the same forward and backwards).

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '81

1.1 A coin is flipped repeatedly. Input the number of heads and the number of tails that occur. Compute the percent of the total flips that are heads and the percent that are tails.

1.2 Find the measure of each angle of a regular polygon, given the number of sides. Sum = 180(n-2)

1.3 Read values for x, A, B, and C and print the value of:

$$Ax^2 + Bx + C.$$

1.4 Often a "chain discount" is applied to an item. First one and then another discount is applied. Given the original price and the two discount rates, compute the final net price of the item.

1.5 Accept a pair of coordinates and determine which Quadrant (I-IV) the point lies in, or if not, which axis.

2.1 Given fractions a/b and c/d (a,b,c, and d integers, b and d not zero), print their sum in lowest terms.

2.2 Given the coordinates of four points in the xy-plane, determine whether the quadrilateral formed by joining the points in order is equilateral.

2.3 Suppose that discounts for direct-dial long-distance calls are figured as follows (where the times are at the calling points).

TIME	DAY	DISCOUNT
7:00 A.M. - 5:00 P.M.	Weekdays	None
5:00 P.M. - 11:00 P.M.	Any day	20%
11:00 P.M. - 7:00 A.M.	Any day	40%
7:00 A.M. - 11:00 P.M.	Saturday	20%
7:00 A.M. - 5:00 P.M.	Sunday	40%

For each call enter the day of the week (1=Sunday, 2=Monday, ... 7=Saturday) and the time the call originated (in terms of a twenty-four hour clock; e.g., for 7:00 P.M. use 1900). Print the discount rate for the call.

2.4 Determine whether the graphs of the equations are parallel. Assume that a straight line may be parallel to itself.

$$Ax + By = C$$
$$Dx + Ey = F$$

2.5 Find the LCM of three integers.

3.1 Given the base ten numeral for a whole number, convert it to a base-n numeral (n is whole number such that n is between 2 and 10 inclusive).

3.2 The MODE of a list of numbers is that value which occurs most often. An array may have more than one mode. Write a program to accept an array and print the mode(s) and the number of times each mode occurs.

3.3 At Tiger Electric Company, the overtime rate for any hours over eight in a day is double the worker's regular rate. Input into an array the worker's employee number and regular pay rate and then the number of hours worked during each of the five weekdays. Compute the employee's gross weekly pay.

3.4 Write a program in which the computer plays tic-tac-toe with a human opponent. Use the following numbering system for the board.

```
 1 | 2 | 3
-----------
 8 | 9 | 4
-----------
 7 | 6 | 5
```

Have the computer always move first to the center square. This reduces the program strategy to offensive moves. An algorithm can be developed in which the computer either wins or ties every game.

3.5 At a certain company the mandatory retirement age is 70. Scan a file of employee's Social Security numbers, names, and birthdates. Print a list of those employees who must retire within the next year, within two years, three, four, and five years. Print the retirement lists in order of birthdays.

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '82

1.1 Write a program to play the following game: The computer picks a number to print the message "I am thinking of a number. What is the number?" The computer tells the user if the guess is too high or too low. Use numbers between 1 and 100 and allow the user 7 guesses.

1.2 Write a program to find all numbers less than 50 which can be written as the sum of two squares, i.e. 20 = 2^2 + 4^2.

1.3 Write a program to determine the sum of all numbers between 100 and 1000 divisible by 14.

1.4 Write a program that will generate a random time (1-12, 0-60) for hours, minutes. Add to this another random time. Have the computer print both times using (:) to separate hours from minutes and the time that the clock would read when added together. (Use only a 12-hour clock).

1.5 Write a program to calculate the roots of a quadratic equation. The equation must be in the following form:
ax^2 + bx + c = 0.

$$\text{roots} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Have the computer print roots if the roots are real. Print only "complex" if not real. The user inputs the coefficients a, b, c.

1.6 Write a program that lists the prime factors of a positive integer <= 100.

1.7 Write a program that calculates the future value of an investment (T) when interest is a factor, using the following formula:

$$T = P(1 + i/N)^{N*y}$$

where T = total value after Years (Future value)
P = initial investment
i = nominal interest rate
N = number of compounding periods per year
y = number of years

1.8 Write a program to find three positive integers such that their sum is 43 and the sum of their cubes is 17299.

1.9 Write a program that prints any keyboard symbol on the screen for exactly 45 seconds.

1.10 Write a program to accept a user's terminating decimal input and print its equivalent fractional notation.

1.11 Write a program that prints an asterisk in the center of the screen. When the user types a U the star moves up, D down, R to the right, L to the left. Do not leave a sketched path, only one asterisk appears on the screen.

2.1 Write a program that calculates the day to the week that a given date falls upon for 1982. January 1 was a Friday. Let the user enter the date in numeric form in the order of the month and day. May 29 will be entered as 5, 29.

2.2 Write a program that calculates the area of a polygon, using the following formula:

$$\text{Area} = [(X1 + X2) * (Y1 + Y2) + (X2 + X3) * (Y2 + Y3) + \\ .... (Xn + X1) * (Yn + Y1)] * 1/2$$

where n = the number of vertices. Let the user supply the x and y coordinates of all vertices. Have the computer ask for the number of vertices.

2.3 Write a program to find a five digit number which when multiplied by four has its digits reversed.

2.4 153 is an interesting number because 153 = 1^3 + 5^3 + 3^3. Write a program to find three other three-digit numbers like 153.

2.5 Write a program to make a computer print the user's name in a zigzag pattern from one side of the screen to the other and back again (only one letter per line.)

2.6 Write a program to repeatedly print a stick figure. Make the figure wave his arm.

2.7 Write a program that will accept the user's letters. Have the program randomly scramble them until a word appears. (The user's letters will be able to make a word).

2.8 Write a program to drill typing skills. Have the computer print four random letters, numbers, or symbols. The computer is to wait for the user's response. If the user types all four correctly, print time that elapsed. If wrong, tell which keys were not typed correctly.

2.9 Write a program that asks the user for the price of an item and what denomination of money the user has. Have the computer print the change returned using the fewest coins and bills possible. (No half-dollars to be used).

2.10 Write a program that converts a measure given in anglo units to metric units. Make the following conversions available: inches to centimeters, feet to centimeters, feet to meters, yards to meters, miles to kilometers.

Note:	1 inch =	2.54 centimeters
	1 foot =	12 inches
	1 yard =	3 feet
	1 meter =	100 centimeters
	1 kilometer =	1000 meters
	1 mile =	5280 feet

2.11 Write a program to find four integers A, B, C and D which satisfy

A^B x C^D = ABCD

2.12 Write a program that calculates the number of days between two given dates in the same year. The program should assume there is one day between today and tomorrow, there are two days between May 1 and May 3.

2.13 Write a program to print a check. Let the user provide the date (using numbers for month, day and year), amount, and payee of the check. The program is to translate the date and amount to words and prints providing spacing within the check.

3.1 Write a program to play Mastermind where the user guesses the computers arrangement of six colors (red, white, blue, green, yellow, black).

3.2 Write a program to plot points on a set of coordinate axes. The user provides the x and y coordinates of all points to be plotted, the end points of the x- and y-axis, and the increment between points on each axis.

The axis are to be switched so that the x-axis runs horizontal. The axes do not necessarily intersect at zero. Where the axes meet is to be printed at the top of each graph.

```
   INTERSECTION AT (50,50)

   * * * * * * * * * * * * * * * * * * * *
   *
   *               + (52, 58)
   *
   *     +
   *
   *           +
   *
   *
   *
   *
```

3.3 A magic square is an array of numbers with just as many rows as columns whereby the sum of any row, column or diagonal is always the same. No number may be used twice in constructing the array.

Write a program to generate magic squares up to 12 x 12. Let the user specify the size desired. The sum in question may be anything.

3.4 Write a program to add and multiply any two Roman numerals inputted. Perform both operations on the numbers and print both answers in Roman form. Have the computer type out the Arabic equivalent underneath each set of equations.

3.5 A 4 digit squumber is a number in which if you add the two left digits to the two right digits you get a number which when squared brings you back to the original number.

```
     example:  2025
               20 + 25 = 45 and (45)^2 = 2025
```

Write a program to find all 4 digit squumbers!

3.6 Write a program to play NIM with a user.

3.7 Write a program that accepts a list of 16 numbers from the user. Have the computer ask the user for another number. Have the computer print between which two numbers in the list the last number falls.

3.8 BONUS

Write a program that guesses the state of the United States of America that the user has in mind. The computer asks yes or no questions.

```
     3 points if guessed in 6 or less questions
     2 points if guessed in 7-9 questions
     1 point  if guessed in 10-12 questions
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '83

1.1 Write a program which, when given any positive decimal number, will round off the number to the nearest whole number and print the results on the screen.

1.2 Write a program which will ask for any five numbers and print out the numbers in descending order.

1.3 Write a program which will print out the factors of any given number which is entered.

1.4 Write a program which, when a name is inputted, will produce a "Happy Birthday Card" (a bordered message with the words Happy and Birthday, and the persons' name centered within the "card").

```
************
*   HAPPY  *
* BIRTHDAY *
*  "Name"  *
************
```

1.5 Write a program which will print a question mark on the screen at different random locations approximately every five seconds.

1.6 Write a program so that when "A" is entered "B" appears on the screen; when "B" is entered "C" appears on the screen. Do this for each letter of the alphabet. A "Z" produces an "A".

1.7 Write a program which will print four distinct rectangles, one in each corner of the screen. Use graphics or a selected character.

1.8 Write a program which will count the number of "e"'s in an inputted sentence.

1.9 Write a program with the following DATA statement:

```
DATA   JOHN,20,70,32
DATA   BILL,71,40,30
DATA   MARY,80,42,73
```

The user will input one of the names and the average of the three test scores will be printed.

1.10 Write a program which will ask for a four digit number and will reverse the digits of the number and print out the results.

2.1 Write a program to calculate the area of a regular hexagon, given the perimeter.

2.2 Write a program to convert a base 8 numeral (at most 4 digits) to a base 2 numeral.

2.3 Write a program to ask the operator for the price of several items. A -999 will terminate the input process. The computer will print the subtotal, tax (5%), and total.

2.4 Write a program to divide the screen into 4 approximately congruent rectangles and allow the operator to pick the character that forms the lines (i.e.; lines made of * or ?, etc.).

2.5 Write a program that will accept a list of numbers. A -999 will terminate the input process. The computer will print the greatest and least in the set.

2.6 Write a program that will accept a list of ten numbers. The computer will print the sum, mean, and median.

2.7 Write a program that will allow an operator to enter a sentence (No period). The computer will reverse the order of the words. Example:

```
THIS IS BACKWARDS
BACKWARDS IS THIS
```

2.8 Write a program that will convert cubic feet to cubic meters. (1 inch = 2.54 cm)

2.9 Write a program that, when given the equation Y = 2(X+5), will find the sum of the Y's when X has consecutive values from a to b. a and b will be single digit numbers entered by the operator.

2.10 Write a program that will print any keyboard character on the screen for 10 seconds, clear the screen, then print the same character twice (two side by side) for 10 seconds, clear the screen and repeat the process 10 times, each time adding one more character.

3.1 Write a program to convert a given number in a given base to any other given base. The bases may vary from 2 - 10.

```
     Sample run:  NUMBER?  112
                  BASE?  3
                  CONVERT TO BASE? 4
                  ANSWER IS  32
```

3.2 Write a program to accept the coordinates of the vertices of a triangle as input. (Collinear points should not be accepted.) The computer will determine if the triangle is equilateral, isosceles, or scalene.

3.3 Write a game program in which the computer secretly and randomly selects one coordinate pair on a 10 X 10 grid. The player inputs a guess. If the guess is right, the game is over. Otherwise the computer gives hints to the player by displaying one of the following messages: UP, DOWN, RIGHT, LEFT, UP AND RIGHT, UP AND LEFT, DOWN AND RIGHT, DOWN AND LEFT.

3.4 Write a program to input two numbers, divide the first number by the second number, and carry the division to a given number of decimal places.

```
     Sample run:    FIRST NUMBER?  2
                    SECOND NUMBER?  7
                    NUMBER OF DECIMAL PLACES?   20
                    QUOTIENT IS .28571428571428571428
```

3.5 The numbers 1 through 8 and a blank are randomly placed in a 3 X 3 array on the screen. For example:

```
                    5  8  3
                    4  1
                    6  2  7
```

Write a program so that when a key from 1 - 8 is pressed, the corresponding number on the screen slides horizontally or vertically to the adjacent blank location, the computer does nothing. (For example: in the above diagram, only the 3, 1, and 7 can be moved.) The user can repeat this process of sliding numbers.

3.6 Write a program that will store a list of spelling words. The program displays a menu with these options:

1. Add a word to the list
2. Delete a word from the list
3. Display the entire list

The program performs the above options.

3.7 Write a program to solve cryptorithms. The user inputs two 2-letter addends and a 3-letter sum. The program will generate a numeric solution if a solution is possible. Only the letters A, B, C, D, and E may be used in the input.

```
Sample run:      FIRST ADDEND?  AB
                 SECOND ADDEND? BC
                 SUM?  DDB

One solution is: A = 5
                 B = 6
                 C = 0
                 D = 1
```

3.8 There are nine lily pads in a row. A frog is sitting on the middle pad. The frog jumps one pad to the right or left with equal probability. Write a program that will collect data on the number of hops a frog takes to land on either one of the two lily pads on the ends.

A simple display should show sequence of jumps that the frog takes.

```
                     F
         -- -- -- -- -- -- -- -- --
```

The frog should repeat the task ten times and the number of hops each time should be displayed on the screen.

```
         10  15  8  4  10  11  17  18  32  6
```

3.9 Write a program in which a user types in a one-line sentence. On the line below the sentence, the cursor can be moved right and left using the R and L keys. If the user presses the space bar, the letter above the cursor is deleted and the rest of the sentence is shifted left.

```
For example:       THIS IS AN EXAMPLE
                    -
becomes:           THS IS AN EXAMPLE
                    -
```

3.10 Write a program that will simulate the movement of a pool ball on a rectangular pool table. The width and length of the table can be any number from 1 to 20. The ball starts at the left hand bottom corner of the table and moves at a 45 degree angle. The ball continues to move until it drops into one of the corner pockets.

The program should display the movement of the ball and indicate which of the three pockets the ball enters. (top-left, top-right, bottom-right).

```
       -----------------           Horizontal = 4
       |  /|\   |  /|\  |           Vertical   = 3
       | / | \  | / | \ |
       |/  |  \|/  |  \|
       --------/--------
       |\  |  /|\  |  /|
       | \ | / | \ | / |
       |  \|/  |  \|/  |
       ----/-------/----
       |  /|\   |  /|\  |
       | / | \  | / | \ |
       |/  |  \|/  |  \|
Starts o---------------o Ends

        Right-bottom
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '84

1.1 Write a program that will produce a table of values in Fahrenheit degrees for given CELSIUS degrees. Have the Celsius table start at 50 and be incremented by 25 stopping at 200. The output should show a two column table with headings CELSIUS on the left and FAHRENHEIT on the right. The values in degrees FAHRENHEIT should be rounded off to the nearest whole number. (The formula to be used is F = 1.8 C + 32)

1.2 Write a program that will compute how long a person sleeps at night in seconds. Assume that the person goes to bed between noon and midnight and gets up between midnight and noon. The computer should ask you to enter the time when you go to bed and the time when you wake up (hours, minutes, and seconds) as follows:

```
     WHAT TIME DID YOU GO TO BED (H, M, S)?
     WHAT TIME DID YOU GET UP (H, M, S)?
```

and then compute:

```
     YOU SLEPT FOR XXXXX SECONDS.
```

1.3 A golfer hits a golf ball in such a way that the horizontal distance from the point of contact to the ball in feet is given by the equation h = 120t, where t is the time in seconds since the ball is hit. The height of the ball above the ground is given by v = 120t - 16t^2. Write a program to produce a three column table, with headings, giving the values of t, h, and v at 1/2 second intervals for values up to the point that the ball returns to the ground. The values of v should be rounded off to the nearest whole number.

1.4 A colony of mice is placed in a closed environment. At the beginning of the experiment there are 10 mice and food supply for 100. Each year the population of mice doubles while the food supply increases only enough to feed 40 more mice. Write a program to print a table of data in the following form:

```
     NUMBER OF YEARS    POPULATION    FOOD SUPPLY FOR

     0                  10            100
     1                  20            140
     .                  .             .
     .                  .             .
     .                  .             .
```

The printing of the table should stop when the population outgrows the food supply.

1.5 Write a program to determine how long, in years, it will take for a savings account to double in value if it is compounded annually. The initial amount of the account as well as the annual interest rate in percent can be entered through input statements.

1.6 Write a program that allows the user to input five first names and then prints (a) the name that is alphabetically closest to the beginning of the alphabet, and (b) closest to the end of the alphabet.

1.7 Write a program to simulate tossing a coin N times and determine the longest run of heads. The program should allow the user to input N, and computer should then print the greatest number of consecutive heads.

1.8 Write a program in which the computer prints the numbers from 1 100, but for any number containing the digit 7, it displays ZAP instead of displaying the number. For a number that contains a 7 and is also divisible by 7 the display should be ZAPZAP.

1.9 Write a program that allows the user to input text for which the computer then prints the number of double letters, i.e. the number of times that a character is the same as the one before it.

1.10 Write a program to assist third-graders with their sevens multiplication facts. In particular, a student should be asked to answer the questions:

```
0 X 7 = ?
1 X 7 = ?
.
.
9 X 7 = ?
```

If a question is answered correctly, the next question should be asked; if not, the question should be repeated. However, if the same question is answered incorrectly twice, the correct answer should be printed and then the next question should be asked.

2.1 Write a program that allows the user to input text for which the computer then prints the total number of vowels that it contains.

2.2 A rational number is the quotient of two integers N/M (M<>0). Write a program that will sort a given sequence of rational numbers and print them out in increasing order. The program should allow the user to input each rational number as a pair of integers (e.g., 3/4 will be input as 3,4). However, the output should be in the form N/M. The user should be able to terminate the input by entering 0,0. Assume the numerators and the denominators do not exceed 99 and no more than 99 rational numbers will be input.

2.3 The number 153 has an interesting property. It equals the sum of the cubes of its digits, that is, 153=1^3 + 5^3 + 3^3. There are only four three-digit numbers (including 153) with this property. Write a program that will generate these four numbers.

2.4 The triangle listed below is generated by following a certain algorithm. Write a program that will generate similar triangles for any number of rows < 20. The program should allow the user to input the number of rows.

```
      1
     232
    34543
   4567654
  567898765
 67890109876
7890123210987
```

2.5 Write a program which generates a page of multiplication exercised similar to those shown below. The program should generate 6 exercises with the factors randomly chosen, and the output should be in the format shown.

```
               MULTIPLICATION DRILL

    1.  53                                4.  12
       X 4                                   X 3
       ----                                  ----

    2.  36                                5.  94
       X 1                                   X 8
       ----                                  ----

    3.  72                                6.  39
       X 6                                   X 5
       ----                                  ----
```

2.6 Write a program that simulates throwing darts at a 5 by 5 square checkerboard. Each throw will hit randomly in any of the 25 possible squares with equal likelihood. N darts are thrown at the board, with N an input value. The results and squares hit are displayed on the screen as below:

```
                      .  .  .  .  .
                      .  .  *  .  *
                      *  *  .  .  .
                      .  .  .  *  .
                      .  .  *  .  .

                      NUMBER OF THROWS = 9
                      NUMBER OF SQUARES HIT = 6
```

In this example an "*" in a position indicates that a square has been hit (at least once), and a "." means the square has not been hit.

2.7 Write a program that allows the user to input a text and then tells the user whether or not it is a palindrome. A palindrome is a statement that reads the same forward or backward after all non-alphabetic characters have been removed.

Example: DOOM AN EVIL DEED, LIVEN A MOOD

2.8 Write a program that will allow a sentence to be input and then print a letter frequency table which will list the frequency and percentage of occurrences of each letter (A through Z) of the text in alphabetical order. The table should be headed with the line

```
     LETTER                FREQUENCY                  PERCENT
```

and should have a final line

```
     TOTAL                   XXX
```

Where XXX is the total number of letters in the sentence. Letters which do not occur in the text should not be listed. Percents should be rounded to the nearest whole percent.

2.9 Write a program that asks the user to type in a sentence and then it prints the longest word in the sentence. Assume that the user always types in a sentence that has one word longer than any of the others.

2.10 The children's game of Rock, Scissors, and Paper is played by two opponents. At each turn both players simultaneously signal their choices of either Rock, Scissors, or Paper. The winner of that turn is determined by the rule that Rock wins over Scissors, Paper wins over Rock, and Scissors wins over Paper. If both choices are the same, it is a tie. Write a program to play a game of Rock, Scissors, Paper between the computer and yourself. At each turn, the computer randomly chooses either Rock, Scissors or Paper (without letting you know). It then prompts you for your choice. You enter R for Rock, S for Scissors, or P for Paper. The program then prints the winner of that turn. The game continues until you enter a Q to quit when it is your turn. The program should print a summary of the game's results.

3.1 Write a program that will begin by displaying an S in the approximate center of the screen. A randomly determined trail of asterisks (*) should then be produced from that point, with the trail continuing until it either leaves the screen or intercepts itself. At this point the trail should stop and the maximum distance that the asterisk has reached from the starting point should be printed. The maximum distance should be the straight lines distance between two points. Upon hitting any key, the program should rerun.

SAMPLE OUTPUT:

```
                        S****
                        *
                        *
                        *
      THE MAXIMUM DISTANCE FROM START IS 5.
```

3.2 Computers are often used in code breaking. In the English language the twelve most frequently used letters are ETAOINSHRDLU. Write a program that will automatically substitute the letter E for the most frequent character in a coded message, the letter T for the second most frequent character, and so on.You may assume that no two characters occur with the same frequency and that no blanks or punctuation occurs. Limit the message to 32 characters.

SAMPLE RUN:

```
             MESSAGE?  ABBRBRXXXX
             THE DECODED MESSAGE IS OTTATAEEEE
```

3.3 The digital product of a positive number N is defined as the product of its non-zero digits. For example, the digital product of the number, 1239, is 54. In turn, the digital product of 54 is 20, and the digital product of 20 is 2. This number 2 is called the digital product root of 1239. The digital product root of number is always a single digit number determined in this fashion. Write a program that requests an input of a number from one to seven digits long and then prints the number followed by the sequence of repeated digital products ending with the digital product root.

SAMPLE RUN:

```
             ORIGINAL VALUE (1 TO 7 DIGITS) ? 9999090
             9999090
             59049
             1620
             12
             2
```

3.4 A positive integer N>1 is called a prime number if its only positive factors are 1 and N. If N and N+2 are both primes, then the two numbers are called "twin primes". Write a program that will request an upper search limit and then display all twin prime pairs not greater than the number set as the upper limit.

SAMPLE RUN:

```
            N= 20
            TWIN PRIME PAIRS NOT GREATER THAN 20

            3     5
            5     7
            11    13
            17    19
```

3.5 There are n people (n<=26) whose names are A,B,C,... and a phone booth with a capacity m (m<=n). Assume that a subset of m people is to be selected from the set of n people to fill the booth. Write a program to list all possible subsets of m people. The program should allow the user to input n and m. The output should list the subsets of names and state the number of subsets.

SAMPLE RUN:

```
            INPUT NUMBER, CAPACITY? 5,3

            ABC ABD ABE ACD ACE
            ADE BCD BCE BDE CDE

            THERE ARE 10 SUBSETS
```

3.6 Write a program to read from DATA statements a collection of strings, and to print a vertical histogram (bar graph) representing the occurrences of each letter of the alphabet (A-Z) in the strings. Assume that the string *END* terminates the data. Characters other than letters are to be ignored. The data will contain no lower case letters.

SAMPLE OUTPUT SHOULD LOOK LIKE:

```
*   *     *
*  **  *  *     *
*  **  *  *  *  *  *
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

3.7 Write a program that will allow the user to input the numerator and denominator of a fraction (N/D) where both numbers are integers 1 to 256. The program should then print the decimal representation indicating those decimals with repeating sequences by enclosing them in parentheses. For example the repeating decimal, .134134134... should be reported as .(134).

SAMPLE OUTPUT:

```
               1/3 = .(3)
               22/5 = 4.4
               3/8 = .375
```

3.8 A positive integer, N, is said to be a "round" number if, when the number is expressed in binary, there is an equal number of 1's and 0's. The number 9 is a round number 26 is not round since its binary representation is 11010. Write a program that, when a positive integer is input, will output the number of round numbers that are less than or equal to the input number.

SAMPLE RUN:

```
             INPUT NUMBER?  10
             THERE ARE 3 ROUND NUMBERS LESS THAN OR EQUAL TO 10
```

3.9 Write a program that will read a series of DATA statements dealing with prices and then will automatically update the money amounts to reflect an input rate of price increase. All money amounts should be rounded to the nearest cent. You will be given a set of DATA lines to run as well as an increase in cost expressed as percent. Assume that the DATA statements will be terminated with *END*.

EXAMPLE:

Assume that your DATA lines are:

```
DATA "THE CURRENT COST OF BUCKLES IS"
DATA "3 FOR $2.50, OR $10.00 A DOZEN."
DATA "*END*"
```

If you input a value of 10% for the price increase, the line would be printed out as:

```
THE CURRENT COST OF BUCKLES IS
3 FOR $2.75, OR $11.00 A DOZEN.
```

3.10 A tennis set between two players has the following rules:

1. A point can be won by either player.
2. A game is won when a player has won at least 4 points and leads the other player by at least 2 points.
3. A set is won when a player has won at least 6 games and leads by at least 2 games.

Write a program that simulates playing several sets of tennis. The two players are designated A and B. The user should be asked to input the number of sets and the probability that player A will win a point. The letter designating the winner of a point is to be printed. At the end of each game the name of the winning player is printed enclosed in parentheses, (A) or (B). At the end of the set, the name of the winner is printed along with the number of wins.

SAMPLE OUTPUT:

```
NUMBER OF SETS = 10
% CHANCE A WINS A POINT=60

AABAAAA (A)
BAAAAABA (A)
AAAABABA (A)
AABABAABA (A)
BABBBAABB (B)
AAABABAA (A)
ABAAABAA (A)
ABAABAAA (A)
AAAAAA (A)
AAAABAA (A)

PLAYER 'A' WON 9 SETS OUT OF 10
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '80
JUDGING CRITERIA

```
1.1  INPUT: Enter number of terms: 9
    OUTPUT: 1  1  2  3  5  8  13  21  34

     INPUT: Enter number of terms: 3
    OUTPUT: 1  1  2


1.2  RUN PROGRAM:                 OUTPUT: (A symbol will flash on
                                          and off approximately
                                          every 10 seconds)


1.3  INPUT: Enter L, W: 10, 8     OUTPUT: 36

     INPUT: Enter L, W: 15, 6     OUTPUT: 42

 1.4  INPUT: Enter C: 100         OUTPUT: 212

     INPUT: Enter C: 0            OUTPUT: 32


1.5  INPUT: Enter character: 5    OUTPUT: NUMERAL

     INPUT: Enter character: R    OUTPUT: NON-NUMERAL

     INPUT: Enter character: 0    OUTPUT: NUMERAL


1.6  INPUT: Enter D, G: 200, 25   OUTPUT: 8

     INPUT: Enter D, G: 240, 15   OUTPUT: 16


1.7  INPUT: Enter number: 30      OUTPUT: YES

     INPUT: Enter number: 99      OUTPUT: NO


1.8  INPUT: Enter L, H: 3, 5      OUTPUT: 4

     INPUT: Enter L, H: 24, 25    OUTPUT: 7


1.9  RUN PROGRAM:                 OUTPUT: (A dot or blob will move
                                          across the screen)
```

```
2.1  INPUT: Enter number of #s: 5         OUTPUT: 9
            Enter #: -3
            Enter #: 0
            Enter #: -8
            Enter #: 9
            Enter #: 4

2.2  INPUT: Enter 4 sides: 4, 5, 6, 0     OUTPUT: TRIANGLE

     INPUT: Enter 4 sides: 3, 3, 3, 3     OUTPUT: SQUARE

     INPUT: Enter 4 sides: 2, 3, 2, 3     OUTPUT: RECTANGLE

2.3  RUN PROGRAM:                         OUTPUT: 1501500

2.4  INPUT: Enter number: 123             OUTPUT: 321

     INPUT: Enter number: 654             OUTPUT: 456

2.5  RUN PROGRAM:                         OUTPUT: (A rectangle
                                                  will be drawn
                                                  somewhere on
                                                  the screen)

2.6  INPUT: Enter 3 numbers: 3, 6, 4      OUTPUT: 3  4  6

     INPUT: Enter 3 numbers: 7, -1, -8    OUTPUT: -8  -1  7

2.7  INPUT: Enter number of #s: 4         OUTPUT: 7
            Enter #: 5
            Enter #: 4
            Enter #: 9
            Enter #: 10

2.8  INPUT: Enter number: 6               OUTPUT: EVEN

     INPUT: Enter number: 27              OUTPUT: ODD
```

```
2.9  INPUT: Enter number: 7               OUTPUT: YES

     INPUT: Enter number: 189             OUTPUT: NO

     INPUT: Enter number: 83              OUTPUT: YES

     INPUT: Enter number: 169             OUTPUT: NO


2.10 INPUT: How many QUARTERS? 3          OUTPUT: 94 CENTS
            How many DIMES? 1
            How many NICKELS? 1
            How many PENNIES? 4

     INPUT: How many QUARTERS? 1          OUTPUT: 53 CENTS
            How many DIMES? 0
            How many NICKELS? 4
            How many PENNIES? 8


2.11 INPUT: Enter sentence: A SENTENCE IS ENTERED.
    OUTPUT: 6

     INPUT: Enter sentence: THIS IS A SHORT ONE.
    OUTPUT: 1
```

```
3.1  INPUT: (Enter the product of two random numbers displayed)

     First, enter the correct answer.          OUTPUT: RIGHT

     On next run enter an incorrect answer.    OUTPUT: WRONG

     On next run enter a correct answer.       OUTPUT: RIGHT


3.2  RUN PROGRAM TWICE:                OUTPUT: (Digits 0-9 will
                                               appear in random
                                               order)


3.3  INPUT: Enter number: 33                   OUTPUT: 30

     INPUT: Enter number: 1098                 OUTPUT: 1100

     INPUT: Enter number: 985                  OUTPUT: 990


3.4  INPUT: Enter number: 83                   OUTPUT: 1103

     INPUT: Enter number: 63                   OUTPUT: 333


3.5  INPUT: Enter number: 122                  OUTPUT: 17

     INPUT: Enter number: 21012                OUTPUT: 194


3.6  INPUT: Enter decimal: .35                 OUTPUT: 7/20
            R or T: T

     INPUT: Enter decimal: .12                 OUTPUT: 2/15
            R or T: R

     INPUT: Enter decimal: .3                  OUTPUT: 1/3
            R or T: R
```

```
3.7  INPUT: Enter cents: 40            OUTPUT: 0 HALF
                                               1 QUARTER
                                               1 DIMES
                                               1 NICKELS
                                               0 PENNIES

     INPUT: Enter cents: 23            OUTPUT: 0 HALF
                                               0 QUARTER
                                               2 DIMES
                                               0 NICKELS
                                               3 PENNIES


3.8  INPUT: Enter guess: 5
    OUTPUT: TOO HIGH     TOO LOW    or    RIGHT ON
     INPUT: Enter guess: (with the output information, continue
            to guess the secret random number [between 1 and 10]
            until output is RIGHT ON)

     RUN PROGRAM AGAIN:


3.9  RUN PROGRAM:                      OUTPUT: A=2  B=5  C=9  D=2


3.10 INPUT: Enter month, day: 6, 3     OUTPUT: TUE

     INPUT: Enter month, day: 12, 31   OUTPUT: WED


3.11 Press appropriate keys to simulate an "etch-a-sketch".
     Press keys to make it go up, right, down, and left.


3.12 INPUT: Enter word: CANNON         OUTPUT: NOT PALINDROME

     INPUT: Enter word: ABBA           OUTPUT: PALINDROME

     INPUT: Enter word: NUN            OUTPUT: PALINDROME
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '81
JUDGING CRITERIA

```
1.1  INPUT: Enter number of heads: 5
            Enter number of tails: 15

    OUTPUT: PERCENT HEADS: 25
            PERCENT TAILS: 75


1.2  INPUT: Enter number of sides: 12

    OUTPUT: ANGLE=150


1.3  INPUT: Enter A, B, C, X: 4, 3, 2, 5

    OUTPUT: AX^2 + BX + C = 117


1.4  INPUT: Enter original price $: 46.00
            Enter discount 1 percent: 10.4
            Enter discount 2 percent: 5

    OUTPUT: FINAL NET PRICE: $39.16


1.5  INPUT: Enter X, Y: 7, -1

    OUTPUT: QUADRANT: IV

     INPUT: Enter X, Y: -9, 0

    OUTPUT: LIES ON THE X-AXIS
```

```
2.1  INPUT: Enter a, b, c, d: 4, 5, 3, 20

   OUTPUT: 19/20
```

```
2.2  INPUT: Enter point 1: 2, 2
            Enter point 2: 3, 5
            Enter point 3: 6, 7
            Enter point 4: 5, 4

    OUTPUT: QUAD IS NOT EQUILATERAL


     INPUT: Enter point 1: 3, 5
            Enter point 2: 1, 1
            Enter point 3: 5, 3
            Enter point 4: 7, 7

    OUTPUT: QUAD IS EQUILATERAL
```

```
2.3  INPUT: Enter day, time: 2, 2100

    OUTPUT: 20%


     INPUT: Enter day, time: 5, 800

    OUTPUT: NO DISCOUNT
```

```
2.4  INPUT: Enter A, B, C: 1, 2, 4
            Enter D, E, F: 2, 4, 0

    OUTPUT: LINES ARE PARALLEL


     INPUT: Enter A, B, C: 1, 3, 6
            Enter D, E, F: 2, 0, 6

    OUTPUT: LINES ARE NOT PARALLEL
```

```
2.5  INPUT: Enter three integers: 12, 15, 8

    OUTPUT: 120
```

```
3.1  INPUT: Enter numeral, base: 576, 5

    OUTPUT: 4301
```

```
3.2  INPUT: Enter how many numbers: 12
            Enter #: 4
            Enter #: 7
            Enter #: 9
            Enter #: 10
            Enter #: 2
            Enter #: 3
            Enter #: 7
            Enter #: 10
            Enter #: 9
            Enter #: 7
            Enter #: 1
            Enter #: 9

    OUTPUT: MODE(S): 7  9
            NUMBER OF OCCURRENCES: 3
```

```
3.3  INPUT: Employee Number: 13
            Regular rate of pay/hour $: 4.50
            Enter hours for M,T,W,R,F: 8, 12, 7, 9, 10

    OUTPUT: EMPLOYEE NUMBER: 13
            GROSS WEEKLY PAY: $238.50
```

3.4 RUN PROGRAM:

Computer geneartes tic-tac-toe board as numbered:

```
 1 | 2 | 3
-----------
 8 | 9 | 4
-----------
 7 | 6 | 5
```

Play tic-tac-toe with computer. It will move in center square first.

Choose a number to make a move. Numbers are replaced by X or O.

```
3.5  INPUT: Enter number of employees: 3

            Social Security No.: 013-42-6900
            Name: MAYE I. RETIRE
            Birthdate (Month and day): JANUARY 20
            Birthdate (Year): 1915

            Social Security No.: 111-11-1111
            Name: RONALD REGAN
            Birthdate (Month and day): APRIL 1
            Birthdate (Year): 1910

            Social Security No.: 222-22-2222
            Name: RICHARD NIXON
            Birthdate (Month and day): JANUARY 30
            Birthdate (Year): 1916

    OUTPUT: RETIRE WITHIN 1 YEARS
              #111-11-1111 RONALD REGAN

            RETIRE WITHIN 2 YEARS
              #111-11-1111 RONALD REGAN

            RETIRE WITHIN 3 YEARS
              #111-11-1111 RONALD REGAN

            RETIRE WITHIN 4 YEARS
              #111-11-1111 RONALD REGAN
              #013-42-6900 MAY I. RETIRE

            RETIRE WITHIN 5 YEARS
              #111-11-1111 RONALD REGAN
              #013-42-6900 MAY I. RETIRE
              #222-22-2222 RICHARD NIXON
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '82
JUDGING CRITERIA

1.1 RUN PROGRAM:
OUTPUT: **I AM THINKING OF A NUMBER. WHAT NUMBER IS IT?**
INPUT: **50**

OUTPUT: **TOO HIGH** then INPUT: **25**
or if
OUTPUT: **TOO LOW** then INPUT: **75**
(Continue to guess in this manner, until you get the #)
(dividing amount by 2 each time)
- Rerun program if you don't guess right in 7 tries.

1.2 RUN PROGRAM:
OUTPUT: **2,5,8,10,13,17,18,20,25,26,29,32,34,37,40,41,45**

1.3 RUN PROGRAM: OUTPUT: **35392**

1.4 RUN PROGRAM: OUTPUT: (2 random clock times and the sum of the 2 times. Example below):

```
 5 : 33
 8 : 44
--------
 2 : 17
```

(run program twice)

1.5 INPUT: Enter a, b, c: **2, 2, 2** OUTPUT: **COMPLEX**

INPUT: Enter a, b, c: **1, -1, -6** OUTPUT: **-2, 3**

1.6 INPUT: Enter number: **42** OUTPUT: **2 3 7**

1.7 INPUT: Enter P,i,N,Y: **100,.01,4,100** OUTPUT: **$271.49**

1.8 RUN PROGRAM: OUTPUT: **7 11 25**

1.9 INPUT: Enter a symbol: *****
OUTPUT: (* is displayed for 45 seconds, then it is cleared)

1.10 INPUT: Enter decimal: **.35** OUTPUT: **7/20**

1.11 RUN PROGRAM: (An asterisk will appear in the center of the screen. Press a **U** to move it up, **D** down, **R** right, **L** to the left. It must not leave a sketched path.)

```
2.1  INPUT: Enter month, day: 12, 31   OUTPUT: FRIDAY

     INPUT: Enter month, day: 5, 29    OUTPUT: SATURDAY
```

```
2.2  INPUT:                                INPUT:
     Enter n: 6                            Enter n: 3
     Enter vertex (X, Y): 5, 1             Enter vertex (X, Y): 1, 1
     Enter vertex (X, Y): 2, 4             Enter vertex (X, Y): 5, 5
     Enter vertex (X, Y): -3, 3            Enter vertex (X, Y): -1, -4
     Enter vertex (X, Y): -3, -2
     Enter vertex (X, Y): -1, -4           OUTPUT: AREA = 6
     Enter vertex (X, Y): 2, -2

    OUTPUT: AREA = 41.5
```

```
2.3  RUN PROGRAM:  OUTPUT: 21978
```

```
2.4  RUN PROGRAM:  OUTPUT: 370  371  407
```

```
2.5  INPUT: Enter name: DOUG WOOLLEY
    OUTPUT: (the following will zigzag from one side of the
            screen to the other and back again)
            D
                 O
                     U
                         G

                                   W
                                       O
                                 O
                            L
                       L
                 E
            Y
```

```
2.6  RUN PROGRAM:  OUTPUT: (A stick figure is repeatedly drawn.
                           It will also wave its arm.)
```

```
2.7  INPUT: How many letters: 4   OUTPUT: (random permutations of
            Enter letter: T               the 4 letters.  Stop
            Enter letter: S               program when you see
            Enter letter: I               LIST as one of them.)
            Enter letter: L
```

```
2.8  RUN PROGRAM:
    OUTPUT: (Four random letters, numbers, or symbols)
     INPUT: (Type all four random characters correctly)
    OUTPUT: (The time elapsed must be printed)

     RUN PROGRAM:
    OUTPUT: (Four random letters, numbers, or symbols).
     INPUT: (Type 2 of the characters incorrectly)
    OUTPUT: (The 2 characters that were typed incorrectly
            must be displayed.  No time is displayed.)
```

```
2.9  INPUT: Enter price $: 11.36
            Enter denomination $: 20

    OUTPUT: 1  $5
            2  DOLLARS
            2  QUARTERS
            1  DIME
            4  PENNIES
```

```
2.10 INPUT: (Choose option to convert inches to centimeters)
            Enter inches: 100
    OUTPUT: 254 CM

     INPUT: (Choose option to convert miles to kilometers)
            Enter miles: 6.2138
    OUTPUT: 10 KM
```

```
2.11 RUN PROGRAM:  OUTPUT: A=2  B=5  C=9  D=2
```

```
2.12 INPUT: Enter Month1, Day1: 4, 8
            Enter Month2, Day2: 10, 16

    OUTPUT: 191 DAYS
```

```
2.13 INPUT: Enter month, day, year: 10, 16, 66
            Enter amount $: 52.13
            Enter payee: DOUG WOOLLEY

    OUTPUT:
      **********************************************************
      *                                         OCT. 16, 1966  *
      *                                                        *
      *   PAY TO THE                                           *
      *   ORDER OF   DOUG WOOLLEY                $52.13        *
      *                                                        *
      *   FIFTY-TWO AND 13/100 DOLLARS                         *
      *                                                        *
      **********************************************************
```

3.1 INPUT: Enter 4 colors from the following six:
(red, white, blue, green, yellow, black)

OUTPUT: The number of colors that are correctly positioned will be indicated by BLACK pegs. The number of colors that are chosen correctly but are not positioned correctly will be indicated by WHITE pegs.

INPUT: Enter another 4 colors according to the rules. Ten attempts are given to guess the computer's 4 colors.

```
3.2  INPUT: Enter end point of x-axis: 30, 34
            Enter end point of y-axis: 20, 50
            Enter increment: 2
            How many points: 2
            Enter point: 24, 40
            Enter point: 26, 44

    OUTPUT: INTERSECTION AT (20,34)
            *********
            *
            *  +
            *    +
            *
            *
```

```
3.3  INPUT: Enter size: 3      OUTPUT: 6 1 8
                                       7 5 3
                                       2 9 4
                                       MAGIC NUMBER = 15
```

(NOTE: Magic number is sum of any row, column, or diagonal, and this number may vary from one team's program to another)

```
     INPUT: Enter size: 4      OUTPUT:  1 15 14  4
                                       12  6  7  9
                                        8 10 11  5
                                       13  3  2 16
                                       MAGIC NUMBER = 34
```

```
3.4  INPUT: Enter Roman Numeral: IV
            Enter Roman Numeral: CDXCVI

    OUTPUT: IV + CDXCVI = D
             4 + 496 = 500

            IV * CDXCVI = MCMLXXXIV
             4 * 496 = 1984
```

```
3.5  RUN PROGRAM:
    OUTPUT: 2025
            3025
            9801
```

3.6 Since the rules are not given with this problem, it is very difficult to judge the program.

```
3.7  INPUT: Enter #: 1
            Enter #: 3
            Enter #: 5
            Enter #: 7
            Enter #: 9
            Enter #: 11
            Enter #: 13
            Enter #: 15
            Enter #: 2
            Enter #: 4
            Enter #: 6
            Enter #: 8
            Enter #: 10
            Enter #: 12
            Enter #: 14
            Enter #: 16
            Enter another number: 2

    OUTPUT: BETWEEN 15 AND 4
```

3.8 BONUS- Choose **FLORIDA**
The computer asks yes or no questions concerning your choice. Count the number of questions it asked before it guessed FLORIDA.

3 points if guessed in 6 or less questions
2 points if guessed in 7-9 questions
1 point if guessed in 10-12 questions

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '83
JUDGING CRITERIA

1.1 INPUT: **1.75**
OUTPUT: **2**

INPUT: **7.2**
OUTPUT: **7**

1.2 INPUT: **5,8,1,3,13** OUTPUT: **13,8,5,3,1**

1.3 INPUT: **24** OUTPUT: **1,24,2,12,3,8,4,6**
(any order)

1.4 INPUT: **SUSAN** OUTPUT:

```
************
*   HAPPY  *
* BIRTHDAY *
*   SUSAN  *
************
```

1.5 RUN PROGRAM:
OUTPUT: (Observe the screen for randomly generated question marks every 5 seconds.)

1.6 INPUT: **C**
OUTPUT: **D**

INPUT: **Z**
OUTPUT: **A**

1.7 RUN PROGRAM:
OUTPUT: (Observe the screen for four rectangles in the corners)

1.8 INPUT: **HELLO THERE** OUTPUT: **3**

1.9 INPUT: **BILL** OUTPUT: **47**

1.10 INPUT: **8375** OUTPUT: **5738**

```
2.1  INPUT: 18                OUTPUT: 23.3827


2.2  INPUT: 372               OUTPUT: 11111010


2.3  INPUT: 1.95              OUTPUT: SUBTOTAL  13.93
             .89                           TAX    .70
             .49                         TOTAL  14.63
           10.60
            -999


2.4  INPUT: *                 OUTPUT:             *
                                                  *
                                                  *
                                                  *
                                       ***********************
                                                  *
                                                  *
                                                  *
                                                  *


2.5  INPUT: 121,45,76,-4,     OUTPUT:  371, -18
            371,.06,3,-18,
            -999


2.6  INPUT: 46,3,8,70,22,     OUTPUT: SUM    262
            46,7,43,9,8               MEAN   26.2
                                      MEDIAN 15.5


2.7  INPUT: ONE TWO THREE FOUR
    OUTPUT: FOUR THREE TWO ONE


2.8  INPUT: 3375
    OUTPUT: 95.5694


2.9  INPUT: 3, 7
    OUTPUT: 100

2.10 INPUT: *
    OUTPUT: (One star for 10 seconds, then two stars for 10
             seconds, etc. until 10 stars for 10 seconds.
             Note: stars side by side.)
```

3.1 INPUT: Number **124** OUTPUT: **57**
Base **6**
Convert to **9**

3.2 INPUT: **-2,-3** OUTPUT: **NOT A TRIANGLE**
0,-2
2,-1

INPUT: **1,3** OUTPUT: **ISOSCELES**
5,5
3,1

3.3 Play the game twice.
Input **4,5**
Follow the directions given by the computer until the game is over.

3.4 INPUT: First Number **1** OUTPUT: **.01234567901234567901**
Second Number **81**
Number of Places **20**

3.5 Press the numbers 1 - 8, and check to see if the appropriate number moves to a blank location. Press a number that is not next to a blank to make sure that it does not move.

3.6 a) Use option 1 to add 2 words to the list.
b) Use option 3 to check to see if the words are added.
c) Similarly delete a word.
d) Use option 3 to see if the word is gone.

3.7 INPUT: First Addend? **AB**
Second Addend? **CD**
Sum? **EBC**

OUTPUT: (Only one of the following may show)

```
A = 3    4    4    6    6    7    7    7    8    8    8    8
B = 2    2    3    3    5    2    5    6    3    4    5    7
C = 9 or 8 or 9 or 7 or 9 or 5 or 8 or 9 or 5 or 6 or 7 or 9
D = 7    6    6    4    4    3    3    3    2    2    2    2
E = 1    1    1    1    1    1    1    1    1    1    1    1
```

INPUT: First Addend? **AA** OUTPUT: **NO SOLUTION POSSIBLE**
Second Addend? **BB**
Sum? **CCC**

3.8 Observe nine lily pads in a row with an "F" first positioned in the middle then bouncing randomly either to the left or the right until it reaches the left end or right end. The number of jumps taken will appear on the screen. This process is repeated 10 times. The numbers displayed must not be less than 4 and most will be less than 32.

```
3.9  INPUT: THIS IS AN EXAMPLE
              -
```

Position the cursor under the first I using the R and L keys. Press the space bar and observe the following:

```
     OUTPUT: THS IS AN EXAMPLE
               -
```

Next, position the cursor under the second space and press the space bar and observe the following:

```
     OUTPUT: THS ISAN EXAMPLE
                   -
```

Next, position the cursor under the first letter by pressing the L key. Press the space bar and observe the following:

```
     OUTPUT: HS ISAN EXAMPLE
             -
```

```
3.10 INPUT: Length: 10
            Width: 8
```

Observe a ball moving from the left-bottom up toward the right-top and bounce off the walls until it reaches the left-top corner. The message "LEFT-TOP" must appear on the screen.

```
     INPUT: Length: 5
            Width: 7
```

Observe a ball moving from the left-bottom up toward the right-top and bounce off the walls until it reaches the right-top corner. The message "RIGHT-TOP" must appear on the screen.

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '84
JUDGING CRITERIA

1.1 RUN PROGRAM: OUTPUT:

```
CELCIUS                     FAHRENHEIT

  50                        122
  75                        167
 100                        212
 125                        257
 150                        302
 175                        347
 200                        392
```

1.2 INPUT: bedtime: **10,45,32**

```
INPUT: bedtime: 10,45,32
       wake up:  7,34,47

OUTPUT: 31755 SECONDS
```

1.3 RUN PROGRAM: OUTPUT:

```
T                H              V
0                0              0
.5               60             56
1                120            104
1.5              180            144
2                240            176
2.5              300            200
3                360            216
3.5              420            224
4                480            224
4.5              540            216
5                600            200
5.5              660            176
6                720            144
6.5              780            104
7                840            56
7.5              900            0
```

1.4 RUN PROGRAM: OUTPUT:

```
NUMBER OF YEARS        POPULATION     FOOD SUPPLY FOR
0                        10               100
1                        20               140
2                        40               180
3                        80               220
4                        160              260
5                        320              300
```

1.5 INPUT: **$500, 9%** INPUT: **$1000, 7%**
OUTPUT: **9 YEARS** OUTPUT: **11 YEARS**

1.6 INPUT: **JANE**
FRANK
ADAM
ARRON
SAMANTHA

OUTPUT: **NAME CLOSEST TO BEGINNING: ADAM**
NAME CLOSEST TO END: SAMANTHA

1.7 INPUT: Have program run 4 times with different values of N entered (for example, 100, 50, 70, 20)

OUTPUT: For each run the longest run of heads could be any number between 3 and 10

1.8 RUN PROGRAM:
A run of the program should contain the numbers from 1 to 100 with the following replacements:

```
 7-ZAPZAP          71-ZAP           87-ZAP
17-ZAP             72-ZAP           97-ZAP
27-ZAP             73-ZAP
37-ZAP             74-ZAP
47-ZAP             75-ZAP
57-ZAP             76-ZAP
67-ZAP             77-ZAPZAP
70-ZAPZAP          78-ZAP
                   79-ZAP
```

1.9 INPUT: **HOORAY FOR HOLLYWOOD**

OUTPUT: **3**

1.10 INPUT: Answer "7 times" questions allowing for all three options: 1-answer correctly,
2-answer first try incorrectly, second try correctly,
3-answer both tries incorrectly

OUTPUT: If a question is answered correctly, the next question should be asked; if not, the question should be repeated. However, if the same question is answered incorrectly twice, the correct answer should be printed and then the next question should be asked.

2.1 INPUT: **HOORAY FOR COMPUTERS**

OUTPUT: **Contains 7 vowels**

2.2 INPUT: Have them place in the following fractions:
2,3, 65,68, 1,2, 3,4, 10,81 0,0

OUTPUT: The fractions should be listed in the following order:

```
10/81
 1/2
 2/3
 3/4
65/68
```

2.3 RUN PROGRAM:

OUTPUT: **153, 370, 371, 407**

2.4 INPUT: Have them input **14**, for 14 rows

OUTPUT:

```
             1
            232
           34543
          4567654
         567898765
        67890109876
       7890123210987
      890123454321098
     90123456765432109
    0123456789876543210
   123456789010987654321
  23456789012321098765432
 3456789012345432109876543
456789012345676543210987654
```

2.5 RUN PROGRAM:

OUTPUT: Have the program run twice. Each time the format should look like the below and the problem set should be different.

```
                MULTIPLICATION DRILL

        1.  53                      4.  12
           X 4                         X 3
           ----                        ----

        2.  36                      5.  94
           X 1                         X 8
           ----                        ----

        3.  72                      6.  39
           X 6                         X 5
           ----                        ----
```

2.6 INPUT: The program should be run twice each for the following values of N; **1,5, and 90.**

OUTPUT: The output should be reasonable given the nature of the program. (SEE DESCRIPTION)
When N=1, only 1 asterisk will appear.
When N=5, either 5 or 4 asterisks show.
When N=90, either 25 or 24 asterisks show.

2.7 INPUT: **GOLF; NO SIR, PREFER PRISON-FLOG**
OUTPUT: **A PALINDROME**

INPUT: **O WHAT A HAPPY LIFE**
OUTPUT: **NOT A PALINDROME**

2.8 INPUT: **HOW ARE YOU?**

OUTPUT:

```
LETTER     FREQUENCY     PERCENT

A          1             11
E          1             11
H          1             11
O          2             22
R          1             11
U          1             11
W          1             11
Y          1             11
TOTAL 9
```

2.9 INPUT: **THE VIOLETS WERE BLUE**

OUTPUT: **VIOLETS**

2.10 RUN PROGRAM:
Run the program for about 10 turns (choose R, S, or P). Look to see if the winner for each turn follows the rules of the game. Choose Q to quit. The summary of the results should be consistent with the results of the individual games. (SEE DESCRIPTION)

3.1 RUN PROGRAM:

Run the program several times. The trail of asterisks will normally be short. When a straight trail is produced, count the number of asterisks, this is the maximum distance. Don't forget that it is possible for the asterisk to move back toward the start before stopping. Under this condition the maximum distance is greater than the final distance.

```
3.2  INPUT: LLLLLLFFFFFGGGGJJJKKB

    OUTPUT: EEEEEETTTTTAAAAOOOIIN
```

```
3.3  INPUT: 54321

    OUTPUT: 54321
            120
            2
```

```
3.4  INPUT: 80

    OUTPUT: 3      5
            5      7
            11    13
            17    19
            29    31
            41    43
            59    61
            71    73
```

```
3.5  INPUT: 7, 4

    OUTPUT: ABCD ABCE ABCF ABCG ABDE ABDF ABDG ABEF ABEG ABFG ACDE
            ACDF ACDG ACEF ACEG ACFG ADEF ADEG ADFG AEFG BCDE BCDF
            BCDG BCEF BCEG BCFG BCEF BDEG BDFG BEFG CDEF CDEG CDFG
            CEFG DEFG

            THERE ARE 35 SUBSETS
```

Different computers will produce different formats, but what counts are the correct combinations and number.

3.6 INPUT: Give the student the following data lines to append to their program:

```
DATA "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG."
DATA "THIS IS AN EXAMPLE OF HOW"
DATA "TO TEST YOUR HISTOGRAM PROGRAM.  YOU"
DATA "CAN USE THIS EXAMPLE."
DATA "*END*"

OUTPUT:                *
             *         *
             *         *
             *         *    *
         *   *         *    *
         *   *  *      *  ***
         *   *  **   * *  ****
         *   *  **   * ** ****
         *   * ***  ***** ****  **
         * * *****  ***** **** ***
         **************************
         ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

3.7 INPUT: **45, 56**

OUTPUT: **45/56 = .803(571428)**

INPUT: **40, 60**

OUTPUT: **40/60 = .(6)**

3.8 INPUT: **100**
OUTPUT: **THERE ARE 14 ROUND NUMBERS LESS THAN OR EQUAL TO 100**

3.9 INPUT: Give the student the following DATA lines:

```
DATA "WE ARE PROUD TO OFFER 4 NEW RECORDS"
DATA "AT A COST OF $1.95 EACH."
DATA "YOU MAY PURCHASE 3 FOR $5.00."
DATA "*END*"
```

INPUT: inflation rate of **6%**

OUTPUT: **WE ARE PROUD TO OFFER 4 NEW RECORDS AT A COST OF $2.07 EACH. YOU MAY PURCHASE 3 FOR $5.30.**

```
3.10  INPUT: NUMBER OF SETS: 10
             % CHANCE A WINS A POINT: 50

     OUTPUT: A Should win about 5 sets

      INPUT: NUMBER OF SETS: 10
             % CHANCE A WINS A POINT: 60

     OUTPUT: A Should win about 9 sets
```

Observe that the scoring rules have been followed and that the correct number of wins has been printed. Check to see that the winner of each set won at least 6 games and led by at least 2 games. (SEE DESCRIPTION)

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '80
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will print terms of the Fibinacci sequence.
'
INPUT "Enter number of terms: "; N
A(1) = 1: A(2) = 1
FOR I = 3 TO N
  A(I) = A(I - 1) + A(I - 2)
NEXT I
FOR I = 1 TO N: PRINT A(I); : NEXT I
PRINT
```

```
'1.2
' This program will flash a symbol on/off every 10 seconds.
'
CLS
FOR I = 1 TO 5
  PRINT "*"
  FOR J = 1 TO 1000: NEXT J
  CLS
  FOR J = 1 TO 10000: NEXT J
NEXT I
```

```
'1.3
' This program will display the perimeter of a rectangle.
'
INPUT "Enter L, W: "; L, W
PRINT L + L + W + W
```

```
'1.4
' This program will convert Celcius to Fahrenheit.
'
INPUT "Enter C: "; C
PRINT (C * 9) / 5 + 32
```

```
'1.5
' This program will determine if input is numeral or not.
'
INPUT "Enter character: "; A$
IF A$ = "0" OR VAL(A$) > 0 THEN PRINT "NUMERAL": END
PRINT "NON-NUMERAL"
```

```
'1.6
' This program will determine gas mileage.
'
INPUT "Enter D, G: "; D, G
PRINT D / G
```

```
'1.7
' This program will test if a number is divisble by 5.
'
INPUT "Enter number: "; N
IF N MOD 5 = 0 THEN PRINT "YES": END
PRINT "NO"
```

```
'1.8
' This program will print the length of a side of a triangle.
'
INPUT "Enter L, H: "; L, H
PRINT SQR(H * H - L * L)
```

```
'1.9
' This program will move a blob across the screen.
'
CLS
FOR C = 1 TO 79
  LOCATE 5, C: PRINT "*"
  FOR J = 1 TO 100: NEXT J
  LOCATE 5, C: PRINT " "
NEXT C
```

```
'2.1
' This program will print the largest number in a sequence.
'
INPUT "Enter number of #s: "; N
L = -999
FOR I = 1 TO N
  INPUT "Enter #: "; X
  IF X > L THEN L = X
NEXT I
PRINT L
```

```
'2.2
' This program will determine what figure is made.
'
INPUT "Enter 4 sides: "; S1, S2, S3, S4
IF S4 = 0 THEN PRINT "TRIANGLE": END
IF S1 = S2 AND S2 = S3 AND S3 = S4 THEN PRINT "SQUARE": END
PRINT "RECTANGLE"
```

```
'2.3
' This program will sum numbers from 1000 to 2000.
'
S = 0
FOR I = 1000 TO 2000
  S = S + I
NEXT I
PRINT S
```

```
'2.4
' This program will reverse a 3 digit number.
'
INPUT "Enter number: "; N$
FOR I = LEN(N$) TO 1 STEP -1
  PRINT MID$(N$, I, 1);
NEXT I
```

```
'2.5
' This program will draw a rectangle on the screen.
'
CLS : SCREEN 1: COLOR 0
LINE (0, 0)-(200, 0): LINE -(200, 100)
LINE -(0, 100): LINE -(0, 0)
LOCATE 23, 1: INPUT A$: SCREEN 0: WIDTH 80
```

```
'2.6
' This program will print 3 numbers in increasing order.
'
INPUT "Enter 3 numbers: "; A(1), A(2), A(3)
FOR I = 1 TO 2
  FOR J = I + 1 TO 3
    IF A(I) > A(J) THEN SWAP A(I), A(J)
  NEXT J
NEXT I
FOR I = 1 TO 3: PRINT A(I); : NEXT I
PRINT
```

```
'2.7
' This program will determine the mean of a set of numbers.
'
INPUT "Enter number of #s: "; N
FOR I = 1 TO N
  INPUT "Enter #: "; X: S = S + X
NEXT I
PRINT S / N
```

```
'2.8
' This program will determine if a number is even or odd.
'
INPUT "Enter number: "; N
IF N MOD 2 = 0 THEN PRINT "EVEN" ELSE PRINT "ODD"
```

```
'2.9
' This program will determine if a number is prime.
'
INPUT "Enter number: ", N
IF N MOD 2 = 0 THEN PRINT "NO": END
FOR I = 3 TO SQR(N)
  IF N MOD I = 0 THEN PRINT "NO": END
NEXT I
PRINT "YES"
```

```
'2.10
' This program will compute value of change.
'
DATA QUARTERS,25,DIMES,10,NICKELS,5,PENNIES,1
FOR I = 1 TO 4
  READ A$(I), A(I): PRINT "How many "; A$(I);
  INPUT X
  S = S + X * A(I)
NEXT I
PRINT S; "CENTS"
```

```
'2.11
' This program will count number of e's in sentence.
'
INPUT "Enter sentence: "; A$
FOR I = 1 TO LEN(A$)
  IF MID$(A$, I, 1) = "E" THEN E = E + 1
NEXT I
PRINT E
```

```
'3.1
' This program allows user to answer multiplication facts.
'
RANDOMIZE TIMER
X = INT(RND(3) * 13): Y = INT(RND(3) * 13)
PRINT X; "X"; Y; " = ";
INPUT N
IF X * Y = N THEN PRINT "RIGHT" ELSE PRINT "WRONG"
```

```
'3.2
' This program will randomize the digits 0..9.
'
RANDOMIZE TIMER
FOR I = 0 TO 9: A(I) = I: NEXT I
FOR I = 0 TO 9
  X = INT(RND(3) * 10)
  Y = A(I): A(I) = A(X): A(X) = Y
NEXT I
FOR I = 0 TO 9: PRINT A(I); : NEXT I: PRINT
```

```
'3.3
' This program will round a number to nearest ten.
'
INPUT "Enter number: "; N
PRINT INT((N + 5) / 10) * 10
```

```
'3.4
' This program will change a number from base 10 to 4.
'
INPUT "Enter number: "; N
J = INT(LOG(N) / LOG(4))
FOR I = J TO 0 STEP -1
  X = INT(N / 4 ^ I): PRINT USING "#"; X;
  N = N - X * 4 ^ I
NEXT I
```

```
'3.5
' This program will change a number from base 3 to 10.
'
INPUT "Enter number: "; N$:
L = LEN(N$)
FOR I = L TO 1 STEP -1
  X = VAL(MID$(N$, I, 1))
  S = S + X * 3 ^ (L - I)
NEXT I
PRINT S
```

```
'3.6
' This program will change a decimal to a fraction.
'
INPUT "Enter decimal: "; N$
INPUT "R or T: "; A$
L = LEN(N$) - 1
N = VAL(RIGHT$(N$, L))
D = INT(10 ^ L + .1)
IF A$ = "R" THEN D = D - INT(D / 10)
FOR I = N TO 1 STEP -1
  IF N MOD I = 0 AND D MOD I = 0 THEN
    PRINT N / I; "/"; D / I: END
  END IF
NEXT I
```

```
'3.7
' This program will represent an amount of money.
'
DATA HALF,50,QUARTER,25,DIMES,10,NICKELS,5,PENNIES,1
INPUT "Enter cents: "; C
FOR I = 1 TO 5
  READ A$, A
  X = INT(C / A): PRINT X; A$
  C = C - X * A
NEXT I
```

```
'3.8
' This program will allow user to guess a generated #.
'
RANDOMIZE TIMER
X = INT(RND(3) * 10) + 1
WHILE X <> G
  INPUT "Enter guess: "; G
  IF G = X THEN PRINT "RIGHT ON": END
  IF G > X THEN PRINT "TOO HIGH" ELSE PRINT "TOO LOW"
WEND
```

```
'3.9
' This program will find fvalues for a,b,c,d.
'
FOR A = 1 TO 9
  FOR B = 0 TO 9
    FOR C = 0 TO 9
      FOR D = 0 TO 9
        N = A * 1000 + B * 100 + C * 10 + D
        M = INT(A ^ B * C ^ D + .1)
        IF N = M THEN
          PRINT "A="; A; " B="; B; " C="; C; " D="; D: END
        END IF
NEXT D, C, B, A
```

```
'3.10
' This program will print day of week for a date.
'
DATA 31,29,31,30,31,30,31,31,30,31,30,31
INPUT "Enter month, day: "; M, D
IF M > 1 THEN
  FOR I = 1 TO M - 1
    READ A: S = S + A
  NEXT I
END IF
S = S + D
X = S - INT(S / 7) * 7
A$ = "MONTUEWEDTHUFRISATSUN"
PRINT MID$(A$, X * 3 + 1, 3)
```

```
'3.11
' This program will simulate an "etch-a-sketch".
'
CLS : ROW = 12: COL = 40
WHILE A$ <> CHR$(27)
  LOCATE ROW, COL: PRINT "*": A$ = ""
  WHILE A$ = "": A$ = INKEY$: WEND
  IF A$ = "I" THEN ROW = ROW - 1
  IF A$ = "M" THEN ROW = ROW + 1
  IF A$ = "J" THEN COL = COL - 1
  IF A$ = "K" THEN COL = COL + 1
WEND
```

```
'3.12
' This program will determine if a word is a palindrome.
'
INPUT "Enter word: "; A$: L = LEN(A$)
FOR I = 1 TO L
  L$ = MID$(A$, I, 1)
  R$ = MID$(A$, L - I + 1, 1)
  IF L$ <> R$ THEN PRINT "NOT "; : I = L
NEXT I
PRINT "PALINDROME"
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '81
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will compute precent of heads and tails.
'
INPUT "Enter number of heads: "; H
INPUT "Enter number of tails: "; T
S = H + T
PRINT "PERCENT HEADS:"; H / S * 100
PRINT "PERCENT TAILS:"; T / S * 100
```

```
'1.2
' This program will display the angle of a polygon.
'
INPUT "Enter number of sides: "; N
PRINT "ANGLE="; 180 * (N - 2) / N
```

```
'1.3
' This program will compute the value of a function.
'
INPUT "A, B, C, X: "; A, B, C, X
PRINT "AX^2 + BX + C ="; A * X * X + B * X + C
```

```
'1.4
' This program will compute the net price after discounts.
'
INPUT "Enter original price: "; P
FOR I = 1 TO 2
  PRINT "Enter discount"; I; " percent:"; : INPUT D
  P = P - P * D / 100
NEXT I
PRINT "FINAL NET PRICE: $"; INT(P * 100 + .5) / 100
```

```
'1.5
' This program will determine the quadrant of a point.
'
INPUT "Enter X, Y: "; X, Y
IF X > 0 AND Y > 0 THEN PRINT "QUADRANT: I"
IF X < 0 AND Y > 0 THEN PRINT "QUADRANT: II"
IF X < 0 AND Y < 0 THEN PRINT "QUADRANT: III"
IF X > 0 AND Y < 0 THEN PRINT "QUADRANT: IV"
IF X = 0 THEN PRINT "LIES ON THE Y-AXIS"
IF Y = 0 THEN PRINT "LIES ON THE X-AXIS"
```

```
'2.1
' This program will sum two fractions.
'
INPUT "Enter a, b, c, d: "; A, B, C, D
NUM = A * D + B * C: DEN = B * D
FOR I = NUM TO 1 STEP -1
  IF NUM MOD I = 0 AND DEN MOD I = 0 THEN
    PRINT NUM / I; "/"; DEN / I: END
  END IF
NEXT I
```

```
'2.2
' This program will determine if quad is equilateral.
'
FOR I = 1 TO 4
  PRINT "Enter point"; I; ":"; : INPUT A(I), B(I)
NEXT I
A(5) = A(1): B(5) = B(1)
FOR I = 1 TO 4
  C(I) = SQR((A(I) - A(I + 1)) ^ 2 + (B(I) - B(I + 1)) ^ 2)
NEXT I
PRINT "QUAD IS ";
FOR I = 1 TO 3
  IF ABS(C(I) - C(I + 1)) > .1 THEN PRINT "NOT EQUILATERAL": END
NEXT I
PRINT "EQUILATERAL"
```

```
'2.3
' This program will print discount rate for phone call.
'
INPUT "Enter day, time: "; D, T
IF T >= 1700 AND T < 2300 THEN PRINT "20%": END
IF T >= 2300 OR T < 700 THEN PRINT "40%": END
IF D = 7 THEN PRINT "20%": END
IF D = 1 THEN PRINT "40%": END
PRINT "NO DISCOUNT"
```

```
'2.4
' This program will determine if graph is parallel.
'
INPUT "Enter A, B, C: "; A, B, C
INPUT "Enter D, E, F: "; D, E, F
PRINT "LINES ARE ";
IF A * E <> D * B THEN PRINT "NOT ";
PRINT "PARALLEL"
```

```
'2.5
' This program will find the LCM of 3 integers.
'
INPUT "Enter three integers: "; A, B, C
FOR I = 1 TO B * C
  S = S + A
  IF S MOD B = 0 AND S MOD C = 0 THEN PRINT S: END
NEXT I
```

```
'3.1
' This program will convert a number from base 10 to 8.
'
INPUT "Enter numeral, base: "; N, B
J = INT(LOG(N) / LOG(B))
FOR I = J TO 0 STEP -1
  X = INT(N / B ^ I): PRINT USING "#"; X;
  N = N - X * B ^ I
NEXT I
PRINT
```

```
'3.2
' This program will print the mode in a list.
'
INPUT "Enter how many numbers: "; N
DIM A(N + 1), B(N + 1): A(N + 1) = -999
FOR I = 1 TO N: INPUT "Enter #: "; A(I): NEXT I
FOR I = 1 TO N
  FOR J = I + 1 TO N
    IF A(I) = A(J) THEN B(I) = B(I) + 1
  NEXT J
  IF B(I) > X THEN X = B(I)
NEXT I
PRINT "MODE(S):";
FOR I = 1 TO N
  IF B(I) = X THEN PRINT A(I);
NEXT I
PRINT : PRINT "NUMBER OF OCCURRENCES:"; X + 1
```

```
'3.3
' This program will compute gross weekly pay.
'
INPUT "Employee Number: "; E$
INPUT "Regular rate of pay/hour $"; R
INPUT "Enter hours for M,T,W,R,F:"; H(1), H(2), H(3), H(4), H(5)
FOR I = 1 TO 5
  IF H(I) <= 8 THEN
    PAY = PAY + H(I) * R
  ELSE
    PAY = PAY + 8 * R + (H(I) - 8) * R * 2
  END IF
NEXT I
PRINT "EMPLOYEE NUMBER: "; E$
PRINT USING "GROSS WEEKLY PAY: $###.##"; PAY
```

```
'3.4
' This program will play tic-tac-toe with a user.
'
'Board numbering system
DATA 1,2,3, 8,9,4, 7,6,5
'Sets of 3 winning squares (in addition to above list)
DATA 1,8,7, 2,9,6, 3,4,5, 1,9,5, 3,9,7
'Vertical and horizontal coordinates for squares
R(1) = 1: C(1) = 1: R(2) = 1: C(2) = 5: R(3) = 1: C(3) = 9
R(4) = 3: C(4) = 9: R(5) = 5: C(5) = 9: R(6) = 5: C(6) = 5
R(7) = 5: C(7) = 1: R(8) = 3: C(8) = 1: R(9) = 3: C(9) = 5
'
RANDOMIZE TIMER
PL$(0) = "YOU": PL$(1) = "COMPUTER"
CLS : A$ = "   |   |": B$ = "---------":
PRINT A$: PRINT B$: PRINT A$: PRINT B$: PRINT A$
FOR I = 1 TO 9
  A(I) = 9: LOCATE R(I), C(I): PRINT MID$(STR$(I), 2, 1)
NEXT I
A(9) = 1: LOCATE R(9), C(9): PRINT "X"
'
FOR MOV = 2 TO 9
  IF MOV = 2 OR MOV = 4 OR MOV = 6 OR MOV = 8 THEN
    P = 0
  ELSE
    P = 1
  END IF
  IF P = 0 THEN
'      Your move
    DO
      LOCATE 8, 3: INPUT "Enter #"; N
      LOCATE 8, 11: PRINT "  "
    LOOP UNTIL A(N) > 1
    A(N) = 0: LOCATE R(N), C(N): PRINT "O"
    GOSUB CheckWinner
  ELSE
'       Computers move
      DO
        X = INT(RND(8) * 8 + 1)
      LOOP UNTIL A(X) > 1
      A(X) = 1: LOCATE R(X), C(X): PRINT "X"
     GOSUB CheckWinner
   END IF
NEXT MOV
LOCATE 10, 3: PRINT "TIE GAME": END
'          Determine if someone wins
CheckWinner:
  FOR I = 1 TO 8
    READ B, C, D
    IF A(B) = P AND A(C) = P AND A(D) = P THEN
      LOCATE 10, 3: PRINT PL$(P); " WON!": END
    END IF
  NEXT I: RESTORE: RETURN
```

```
'3.5
' This program will print a list of people who will retire.
'
TM = 4: TY = 1981    'Today's month and year
INPUT "Enter number of employees:"; N
FOR I = 1 TO N
  PRINT
  INPUT "Social Security No.:"; S$(I)
  INPUT "Name:"; N$(I)
  INPUT "Birthdate (Month and day:"; BM$(I)
  INPUT "Birthdate (Year):"; BY(I)
NEXT I
'          Determine who retires when
FOR Y = TY - 69 TO TY - 65
  YR = Y - (TY - 70)
  FOR I = 1 TO N
    IF BY(I) <= Y THEN
      A(YR) = A(YR) + 1
      A$(YR, A(YR)) = "  #" + S$(I) + " " + N$(I)
      B(YR, A(YR)) = BY(I)
    END IF
  NEXT I
NEXT Y
'          Display retirers
FOR Y = 1 TO 5
  IF A(Y) > 0 THEN
    GOSUB SortDates
    PRINT : PRINT "RETIRE WITHIN"; Y; "YEARS"
    FOR I = 1 TO A(Y): PRINT A$(Y, I): NEXT I
  END IF
NEXT Y: END
'          Sort people by birthdates
SortDates:
  FOR I = 1 TO A(Y) - 1
    FOR J = I + 1 TO A(Y)
      IF B(Y, I) > B(Y, J) THEN
        X = B(Y, I): B(Y, I) = B(Y, J): B(Y, J) = X
        X$ = A$(Y, I): A$(Y, I) = A$(Y, J): A$(Y, J) = X$
      END IF
    NEXT J
  NEXT I: RETURN
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '82
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will allow a user to guess a generated #.
'
RANDOMIZE TIMER
X = INT(RND(3) * 100) + 1:  I = 1
WHILE (I <= 7) AND (G <> X)
  INPUT "I AM THINKING OF A NUMBER. WHAT IS IT"; G
  IF G < X THEN PRINT "TOO LOW"
  IF G > X THEN PRINT "TOO HIGH"
  IF G = X THEN PRINT "RIGHT"
  I = I + 1
WEND
PRINT "NUMBER IS "; X
```

```
'1.2
' This program will find #s that are the sum of 2 squares.
'
DIM A(100)
FOR I = 1 TO 5
  FOR J = I TO 7
    A(I * I + J * J) = -1
  NEXT J
NEXT I
FOR I = 1 TO 49
  IF A(I) THEN PRINT LTRIM$(STR$(I)); ",";
NEXT I: PRINT
```

```
'1.3
' This program will sum numbers divisible by 14.
'
FOR I = 100 TO 1000
  IF I MOD 14 = 0 THEN S = S + I
NEXT I
PRINT S
```

```
'1.4
' This program will add 2 random clock times.
'
RANDOMIZE TIMER
FOR I = 1 TO 2
  H(I) = INT(RND(3) * 12) + 1
  M(I) = INT(RND(3) * 60)
  IF H(I) < 10 THEN PRINT " ";
  PRINT H(I); ": ";
  IF M(I) < 10 THEN PRINT " 0";
  PRINT LTRIM$(STR$(M(I)))
NEXT I
PRINT "--------"
M = M(1) + M(2)
IF M > 59 THEN M = M - 60: H = 1
H = H + H(1) + H(2): IF H > 12 THEN H = H - 12
IF H < 10 THEN PRINT " ";
PRINT H; ": ";
IF M < 10 THEN PRINT "0";
PRINT LTRIM$(STR$(M)): PRINT
```

```
'1.5
' This program will compute roots of equation.
'
INPUT "Enter a, b, c: "; A, B, C
S = B * B - 4 * A * C
IF S < 0 THEN PRINT "COMPLEX": END
PRINT (-B + SQR(S)) / (2 * A);
PRINT (-B - SQR(S)) / (2 * A)
```

```
'1.6
' This program will print prime factors.
'
INPUT "Enter number: "; N
FOR I = 2 TO N
  IF N MOD I = 0 THEN
    IF I = 2 THEN
      PRINT 2;
    ELSE
      J = 1
      DO
        J = J + 1
      LOOP UNTIL J > SQR(I) OR (I MOD J = 0)
      IF I MOD J > 0 THEN PRINT I;
    END IF
  END IF
NEXT I: PRINT
```

```
'1.7
' This program will calculate future value of investment.
'
INPUT "Enter P, i, N, Y: "; P, I, N, Y
' P * (1 + I/N)^(N*Y)
FOR J = 1 TO N * Y
  P = P + P * I / N
NEXT J
PRINT "$"; INT(P * 100 + .5) / 100
```

```
'1.8
' This program will find 3 #s whose sum is 43.
'
FOR I = 1 TO 41
  FOR J = 1 TO 42 - I
    K = 43 - I - J
    IF I * I * I + J * J * J + K * K * K = 17299 THEN
      PRINT I; J; K: END
    END IF
  NEXT J
NEXT I
```

```
'1.9
' This program will print a symbol for 45 seconds.
'
INPUT "Enter a symbol: "; A$
CLS : PRINT A$
FOR I = 1 TO 45
   FOR J = 1 TO 3600: NEXT J  'About one second on 386
NEXT I
CLS
```

```
'1.10
' This program will convert decimal to fraction
'
INPUT "Enter decimal: "; N$
L = LEN(N$) - 1
N = VAL(RIGHT$(N$, L))
D = 1
FOR I = 1 TO L: D = D * 10: NEXT I
FOR I = N TO 1 STEP -1
  IF N MOD I = 0 AND D MOD I = 0 THEN
    PRINT N / I; "/"; D / I: END
  END IF
NEXT I
```

```
'1.11
' This program will move an asterisk by pressing keys.
'
CLS : R = 10: C = 40
LOCATE R, C: PRINT "*"
WHILE A$ <> " "
  DO: A$ = INKEY$: LOOP UNTIL A$ <> ""
  CLS
  IF A$ = "U" THEN R = R - 1
  IF A$ = "D" THEN R = R + 1
  IF A$ = "L" THEN C = C - 1
  IF A$ = "R" THEN C = C + 1
  LOCATE R, C: PRINT "*"
WEND
```

```
'2.1
' This program will print day of week of a date.
' January 1 was a Friday (in 1982).
'
DIM M(12)
DATA 31,28,31,30,31,30,31,31,30,31,30,31
FOR I = 1 TO 12: READ M(I): NEXT I
INPUT "Enter month, day: "; MO, DA
IF MO > 1 THEN
  FOR I = 1 TO MO - 1: S = S + M(I): NEXT I
END IF
S = S + DA
X = S - INT(S / 7) * 7
A$ = "THUFRISATSUNMONTUEWED"
PRINT MID$(A$, X * 3 + 1, 3)
```

```
'2.2
' This program will calculate the area of a polygon.
'
INPUT "Enter n: "; N
FOR I = 1 TO N
  INPUT "Enter vertex (X, Y): "; X(I), Y(I)
NEXT I
X(N + 1) = X(1): Y(N + 1) = Y(1)
FOR I = 1 TO N
  SUM = SUM + X(I) * Y(I + 1) - Y(I) * X(I + 1)
NEXT I
PRINT "AREA ="; ABS(SUM) / 2
```

```
'2.3
' This program will find 5 digit number.
'Strategy: # is less than 25000 because 4 * # would be
'          a 6 digit # otherwise.
'          # can't be 1XXXY since 4 * Y can't give us
'          a 1 in the units place.
'          # must therefore begin with 2 and end with
'          8 since 4*8 = 32.  So we can step 10.
'
FOR I = 20008 TO 24998 STEP 10
  N$ = MID$(STR$(I), 2, 5):  S$ = MID$(STR$(I * 4), 2, 5)
  J = 0
  DO
    J = J + 1
  LOOP UNTIL (J = 5) OR (MID$(N$, J, 1) <> MID$(S$, 6 - J, 1))
  IF MID$(N$, J, 1) = MID$(S$, 6 - J, 1) THEN PRINT I: END
NEXT I
```

```
'2.4
' This program will find interesting numbers.
'
FOR I = 1 TO 9
  FOR J = 0 TO 9
    FOR K = 0 TO 9
      NUM = I * 100 + J * 10 + K
      POW = I * I * I + J * J * J + K * K * K
      IF (NUM = POW) AND (NUM <> 153) THEN
        PRINT I * 100 + J * 10 + K;
      END IF
    NEXT K
  NEXT J
NEXT I: PRINT
```

```
'2.5
' This program will make user's name zigzag.
'
INPUT "Enter name: "; NAM$:  L = LEN(NAM$)
CLS
X = INT(159 / (L - 1))
FOR I = 1 TO L
  M$ = MID$(NAM$, I, 1)
  S = (I - 1) * X: IF S > 79 THEN S = 159 - S
  PRINT TAB(S); M$
NEXT I
```

```
'2.6
' This program will print a stick figure.
'
R = 5: C = 12
DO
  FOR I = 0 TO 5
    CLS
    PRINT " *     ***** "
    PRINT "  *    *   * "
    PRINT "   *   ***** "
    PRINT "    **   *   "
    PRINT "      *****  "
    PRINT "         *   "
    PRINT "         *   "
    PRINT "        * *  "
    PRINT "       *   * "
    PRINT "      *     *"
    INC = (R - I) / 7
    FOR K = 0 TO 6
      LOCATE R - INC * K, C + K: PRINT "*"
    NEXT K
  NEXT I
  A$ = INKEY$
LOOP UNTIL A$ = CHR$(27)
```

```
'2.7
' This program will display permutations of letters.
'
RANDOMIZE TIMER
INPUT "How many letters: "; N
FOR I = 1 TO N: INPUT "Enter letter: "; A$(I): NEXT I
DO
  FOR I = 1 TO N
    X = INT(RND(5) * N + 1)
    T$ = A$(X): A$(X) = A$(I): A$(I) = T$
  NEXT I
  FOR I = 1 TO N: PRINT A$(I); : NEXT I: PRINT
  A$ = INKEY$
LOOP UNTIL A$ = CHR$(27)
```

```
'2.8
' This program will drill typing skills.
'
RANDOMIZE TIMER
FOR I = 1 TO 4
  X = INT(RND(3) * 58) + 33
  A$(I) = CHR$(X): PRINT A$(I); " ";
NEXT I: PRINT : J = 1
WHILE J < 5
   DO: S = S + 1: A$ = INKEY$: LOOP UNTIL A$ <> ""
   B$(J) = A$: PRINT B$(J); " ";
   J = J + 1
WEND
'
PRINT : PRINT
FOR I = 1 TO 4
  IF A$(I) <> B$(I) THEN PRINT A$(I), B$(I), "NO": C = 1
NEXT I
IF C = 0 THEN PRINT INT(S / 3000); " SECONDS"
```

```
'2.9
' This program will return change in fewest coins.
'
DATA $20,2000,$10,1000,$5,500,DOLLARS,100
DATA QUARTERS,25,DIMES,10,NICKELS,5,PENNIES,1
INPUT "Enter price $:"; P
INPUT "Enter denomination $:"; D
N = (D - P) * 100
FOR I = 1 TO 8
  READ A$, A
  X = INT(N / A)
  IF X > 0 THEN PRINT X; " "; A$
  N = INT(N - X * A + .01)
NEXT I
```

```
'2.10
' This program will make unit conversions.
'
DATA IN,CM,FT,CM,FT,M,YD,M,MI,KM
FOR I = 1 TO 5
  READ A$, B$: PRINT I; "  "; A$; "-> "; B$
NEXT I: RESTORE
INPUT "Enter Choice #: "; X
FOR I = 1 TO X: READ A$, B$: NEXT I
PRINT "Enter "; A$; : INPUT N
S = N * 2.54
IF X = 1 THEN PRINT S;
IF X = 2 THEN PRINT S * 12;
IF X = 3 THEN PRINT S * 12 / 100;
IF X = 4 THEN PRINT S * 36 / 100;
IF X = 5 THEN PRINT S * 5280 * 12 / 100000!;
PRINT " "; B$
```

```
'2.11
' This program will find A^B x C^D = ABCD.
'
FOR A = 1 TO 9
  FOR B = 0 TO 9
    FOR C = 0 TO 9
      FOR D = 0 TO 9
        APOW = 1: CPOW = 1
        FOR J = 1 TO B: APOW = APOW * A: NEXT J
        FOR J = 1 TO D: CPOW = CPOW * C: NEXT J
        NUM = A * 1000 + B * 100 + C * 10 + D
        IF APOW * CPOW = NUM THEN
          PRINT "A="; A; " B="; B; " C="; C; " D="; D: END
        END IF
      NEXT D
    NEXT C
  NEXT B
NEXT A
```

```
'2.12
' This program calculates days between 2 dates.
'
DATA 31,28,31,30,31,30,31,31,30,31,30,31
INPUT "Enter Month1, Day1: "; M1, D1
INPUT "Enter Month2, Day2: "; M2, D2
M = M2 - M1
IF M > 0 THEN
  IF M1 > 1 THEN
    FOR I = 1 TO M1 - 1: READ D: NEXT I
  END IF
  FOR I = M1 TO M2 - 1
    READ D: S = S + D
  NEXT I
END IF
PRINT S + D2 - D1; "DAYS"
```

```
'2.13
' This program will print a check.
'
DATA JAN.,FEB.,MAR.,APRIL,MAY,JUNE,JULY,AUG.,SEPT.
DATA OCT.,NOV.,DEC.
DATA ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE
DATA TEN,ELEVEN,TWELVE,THIRTEEN,FOURTEEN,FIFTEEN
DATA SIXTEEN,SEVENTEEN,EIGHTEEN,NINETEEN
DATA TWENTY-,THIRTY-,FOURTY-,FIFTY-,SIXTY-,
DATA SEVENTY-,EIGHTY-,NINETY-
INPUT "Enter month, day, year: ", M, D$, Y$
INPUT "Enter amount $"; N
INPUT "Enter payee: "; N$
'    Display check border
CLS
FOR I = 1 TO 60: PRINT "*"; : NEXT I: PRINT
FOR I = 1 TO 7
  LOCATE I + 1, 1: PRINT "*": LOCATE I + 1, 60: PRINT "*"
NEXT I
FOR I = 1 TO 60: PRINT "*"; : NEXT I: PRINT
'    Display date
FOR I = 1 TO M: READ M$: NEXT I
LOCATE 2, 45: PRINT M$; " "; D$; ", 19"; Y$
IF M < 12 THEN
  FOR I = M + 1 TO 12: READ M$: NEXT I
END IF
'    Display Name and amount
LOCATE 4, 5: PRINT "PAY TO THE"
LOCATE 5, 5: PRINT "ORDER OF  "; N$
LOCATE 5, 50: PRINT "$"; N
LOCATE 7, 3
'    Display amount in words
CENT = INT((N - INT(N)) * 100 + .01): S = 1000
FOR I = 2 TO 0 STEP -1
  S = S / 10:  X = INT(N / S + .001)
  IF (I = 2) AND (X > 0) THEN
    FOR J = 1 TO X: READ E$: NEXT J
    PRINT E$; " HUNDRED ";
  END IF
  IF (I = 1) AND (X > 1) THEN
    FOR J = 1 TO 18 + X: READ E$: NEXT J
    PRINT E$;
  END IF
  IF I = 1 AND X = 1 THEN T = 1
  IF I = 0 THEN
    FOR J = 1 TO T * 10 + X: READ E$: NEXT
    PRINT E$;
  END IF
  RESTORE: FOR K = 1 TO 12: READ X$: NEXT K
  N = INT(N - X * S + .001)
NEXT I
PRINT " AND"; CENT; "/100 DOLLARS"
```

```
'3.1
' This program  will play mastermind.
'
RANDOMIZE TIMER
CLS : LOCATE 1, 5: PRINT "GUESS: W, Y, R, G, BL, BK"
DATA W,Y,R,G,BL,BK
'   Assign random colors
FOR I = 1 TO 4
  X = INT(RND(6) * 6) + 1
  FOR J = 1 TO X: READ A$(I): NEXT J: RESTORE
NEXT I
'    Allow user 10 sets of guesses
FOR K = 1 TO 10
  W = 0: B = 0
  FOR I = 1 TO 4: LOCATE K * 2, I * 6: INPUT B$(I): NEXT I
  FOR I = 1 TO 4: C$(I) = A$(I): NEXT I
  FOR I = 1 TO 4
    IF C$(I) = B$(I) THEN B = B + 1: B$(I) = "": C$(I) = " "
  NEXT I
  FOR I = 1 TO 4
    FOR J = 1 TO 4
       IF C$(I) = B$(J) THEN W = W + 1: B$(J) = "": C$(I) = " "
    NEXT J
  NEXT I
'      Black pegs = Correct color and correct position
'      White pegs = Correct color but wrong position
  LOCATE K * 2, 40: PRINT "BLACKS = "; B; "   WHITES ="; W
  IF B = 4 THEN PRINT "YOU WIN IN "; K; " TURNS": END
NEXT K
PRINT "YOU LOSE"
FOR I = 1 TO 4: PRINT A$(I); " "; : NEXT I

'3.2
' This program will plot points on a new axis.
'
CLS
INPUT "Enter end point of x-axis: "; X1, Y1
INPUT "Enter end point of y-axis: "; X2, Y2
INPUT "Enter increment: "; IT
INPUT "How many points: "; N
FOR I = 1 TO N
  INPUT "Enter point: "; X(I), Y(I)
NEXT I
CLS : R = 3: C = 1
PRINT "INTERSECTION AT ("; X2; ","; Y1; ")"
PRINT
FOR I = Y1 TO Y2 STEP IT:   PRINT "*"; : NEXT I: PRINT
FOR I = X2 + 1 TO X1 STEP IT: PRINT "*": NEXT I
FOR I = 1 TO N
  LOCATE R + (X(I) - X2) / IT, C + (Y(I) - Y1) / IT: PRINT "+"
NEXT I
```

```
'3.3
' This program will generate magic squares.
' -- Correct for odd sizes and 4x4.
'
CLS
INPUT "Enter size: "; N:  DIM A(N, N): PRINT
IF N MOD 2 > 0 THEN
  X = 1: Y = (N + 1) / 2: A(X, Y) = 1
'   Routine for Odd matrix
  FOR I = 2 TO N * N
    X = X - 1: Y = Y - 1
    IF X = 0 THEN X = N
    IF Y = 0 THEN Y = N
    IF A(X, Y) = 0 THEN
      A(X, Y) = I
    ELSE
      X = X + 2: Y = Y + 1
      IF X > N THEN X = X - N
      IF Y > N THEN Y = 1
      A(X, Y) = I
    END IF
  NEXT I
ELSE
'  Routine for Even matrix
  FOR I = 1 TO N
    FOR J = 1 TO N
      S = S + 1
      IF I = J OR I = N + 1 - J THEN
        A(I, J) = S
      ELSE
        A(I, J) = N * N + 1 - S
      END IF
   NEXT J
 NEXT I
END IF
FOR I = 1 TO N
  FOR J = 1 TO N
    LOCATE I * 2, J * 4: PRINT A(I, J)
  NEXT J
NEXT I
PRINT : PRINT "MAGIC NUMBER="; (N * N * N + N) / 2
```

```
'3.4
' This program will add and multiply 2 Roman Numerals.
'
DATA M,1000,D,500,C,100,L,50,X,10,V,5,I,1
FOR I = 1 TO 7: READ RN$(I), RV(I): NEXT I
FOR E = 1 TO 2
  INPUT "Enter Roman Numeral: "; ROM$(E)
  L = LEN(ROM$(E)): I = 1: AR = 0
  WHILE I < L
    C$ = MID$(ROM$(E), I, 1)
    I1 = 1: WHILE C$ <> RN$(I1): I1 = I1 + 1: WEND
    NC$ = MID$(ROM$(E), I + 1, 1)
     I2 = 1: WHILE NC$ <> RN$(I2): I2 = I2 + 1: WEND
     IF I1 <= I2 THEN
       AR = AR + RV(I1)
     ELSE
       AR = AR + RV(I2) - RV(I1): I = I + 1
     END IF
     I = I + 1
   WEND
   IF I <= L THEN
'       Last numeral was not done
     C$ = MID$(ROM$(E), I, 1)
     I1 = 1: WHILE C$ <> RN$(I1): I1 = I1 + 1: WEND
     AR = AR + RV(I1)
   END IF
   A(E) = AR
NEXT E
'    Convert Arabic numbers to Roman Numerals
N(1) = A(1) + A(2): N(2) = A(1) * A(2)
FOR K = 1 TO 2: NUM = N(K)
  FOR I = 1 TO 7
    X = NUM / RV(I)
    BOOL = (X < 2) AND (X >= 9 / 5)
    BOOL = BOOL AND ((I = 2) OR (I = 4) OR (I = 6))
    IF NOT BOOL THEN
      XX = INT(X)
      IF XX = 9 THEN
        R$(K) = R$(K) + RN$(I) + RN$(I - 2)
      ELSEIF XX = 4 THEN
        R$(K) = R$(K) + RN$(I) + RN$(I - 1)
      ELSEIF XX > 0 THEN
        FOR J = 1 TO XX: R$(K) = R$(K) + RN$(I): NEXT
      END IF
      NUM = NUM - RV(I) * XX
    END IF
  NEXT I
NEXT K
'     Display sum and product
PRINT ROM$(1); " + "; ROM$(2); " = "; R$(1)
PRINT A(1); "+"; A(2); "="; N(1)
PRINT
PRINT ROM$(1); " * "; ROM$(2); " = "; R$(2)
PRINT A(1); "*"; A(2); "="; N(2)
```

```
'3.5
' This program will find 4 digit squumbers.
'
FOR I = 1000 TO 9999
  L = INT(I / 100)
  R = I MOD 100
  X = L + R
  IF X * X = I THEN PRINT I
NEXT I

'3.6
' Write a program to play NIM with a user.
' Since the rules are not given with this problem,
' it is very difficult to write the program.


'3.7
' This program will determine where a # falls in a list.
'
DIM A(16)
FOR I = 1 TO 16
  INPUT "Enter #: "; A(I)
NEXT I
INPUT "Enter another number: "; NUM
I = 1
WHILE A(I) <> NUM: I = I + 1: WEND
PRINT "BETWEEN"; A(I - 1); "AND"; A(I + 1)


'3.8
' This BONUS program will guess the user's state.
'
DATA ALABAMA,ALASKA,ARIZONA,ARKANSAS,CALIFORNIA
DATA COLORADO,CONNECTICUT,DELEWARE,FLORIDA,GEORGIA
DATA HAWAII,IDAHO,ILLINOIS,INDIANA,IOWA,KANSAS
DATA KENTUCKY,LOUISIANA,MAINE,MARYLAND,MASSACHUSETTS
DATA MICHIGAN,MINNESOTA,MISSISSIPPI,MISSOURI,MONTANA
DATA NEBRASKA,NEVADA,NEW HAMPSHIRE,NEW JERSEY,NEW YORK
DATA NEW MEXICO,NORTH CAROLINA,NORTH DAKOTA,OHIO,OKLAHOMA
DATA OREGON,SOUTH CAROLINA,SOUTH DAKOTA,PENNSYLVANIA
DATA RHODE ISLAND,TENNESSEE,TEXAS,UTAH,VERMONT,VIRGINIA
DATA WASHINGTON,WEST VIRGINIA,WISCONSIN,WYOMING
DIM ST$(50)
FOR I = 1 TO 50: READ ST$(I): NEXT I
G = 1: B = 1: M = 25: E = 50
DO
  PRINT G; "- IS YOUR STATE ALPHABETICALLY BEFORE "; ST$(M)
  INPUT "Enter YES or NO: "; A$
  IF A$ = "YES" AND B + 1 = M THEN PRINT ST$(B); " IS IT": END
  IF A$ = "NO" AND M = E THEN PRINT ST$(M); " IS IT": END
  IF A$ = "YES" THEN E = M - 1: M = M - INT((M - B) / 2 + .5)
  IF A$ = "NO" THEN B = M:  M = M + INT((E - M) / 2 + .5)
  G = G + 1
LOOP UNTIL G > 12
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '83
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will round a number to nearest whole number.
'
INPUT "Enter number: "; N
PRINT INT(N + .5)


'1.2
' This program will display 5 numbers in descending order.
'
FOR I = 1 TO 5
  INPUT "Enter number: "; A(I)
NEXT I
FOR I = 1 TO 4
  FOR J = I + 1 TO 5
    IF A(I) < A(J) THEN SWAP A(I), A(J)
  NEXT J
NEXT I
FOR I = 1 TO 5: PRINT A(I): NEXT I


'1.3
' This program will print the factors of a given number.
'
INPUT "Enter number: "; N
FOR I = 1 TO N
  IF N MOD I = 0 THEN PRINT I
NEXT I


'1.4
' This program will produce a birthday card w/name centered.
'
INPUT "Enter name: "; N$
FOR I = 1 TO 5
  PRINT
  IF I = 1 OR I = 5 THEN FOR J = 1 TO 12: PRINT "*"; : NEXT J
  IF I = 2 THEN PRINT "*   HAPPY  *";
  IF I = 3 THEN PRINT "* BIRTHDAY *";
  IF I = 4 THEN
    PRINT "*"; : L = LEN(N$)
    SP = INT((10 - L) / 2 + .5)
   PRINT SPACE$(SP); N$; SPACE$(10 - L - SP); "*";
  END IF
NEXT I
```

```
'1.5
' This program will print a ? in random locations.
'
CLS : RANDOMIZE TIMER
FOR I = 1 TO 6
  C = INT(RND(3) * 80) + 1:  R = INT(RND(3) * 23) + 1
  LOCATE R, C: PRINT "?";
  FOR J = 1 TO 3000 * 5: NEXT J
NEXT I
```

```
'1.6
' This program will print a B for a, C for B, ... Z for A.
'
INPUT "Enter character: "; CH$
IF CH$ < "Z" THEN PRINT CHR$(ASC(CH$) + 1) ELSE PRINT "A"
```

```
'1.7
' This program will print 4 distinct rectangles in corners.
'
CLS
R = 1: C = 1:   GOSUB DrawRec
R = 1: C = 65:  GOSUB DrawRec
R = 19: C = 1:  GOSUB DrawRec
R = 19: C = 65: GOSUB DrawRec
END
DrawRec:
  FOR I = R TO R + 3
    IF I <> R + 1 AND I <> R + 2 THEN
      LOCATE I, C
      FOR J = 1 TO 10: PRINT "*"; : NEXT J
    END IF
    LOCATE I, C: PRINT "*"
    LOCATE I, C + 9: PRINT "*"
  NEXT I
RETURN
```

```
'1.8
' This program will count the number of E's in a sentence.
'
INPUT "Enter sentence: "; S$
FOR I = 1 TO LEN(S$)
  C$ = MID$(S$, I, 1)
  IF C$ = "E" THEN E = E + 1
NEXT I
PRINT E
```

```
'1.9
' This program will calculate the average socre for a person.
'
DATA JOHN,20,70,32
DATA BILL,71,40,30
DATA MARY,80,42,73
INPUT "Enter name: "; N$
FOR I = 1 TO 3
  READ B$, A, B, C
  IF B$ = N$ THEN PRINT (A + B + C) / 3
NEXT I
```

```
'1.10
' This program will reverse the digits of a 4 digit number.
'
INPUT "Enter number: "; N$
FOR I = 4 TO 1 STEP -1
  PRINT MID$(N$, I, 1);
NEXT I
```

```
'2.1
' This program will calculate the area of a regular hexagon.
'
INPUT "Enter perimeter: "; P
S = P / 6
PRINT SQR(3) * S / 2 * S / 2 * 6
```

```
'2.2
' This program will convert a base 8 num to a base 2 num.
'
INPUT "Enter number: "; N$:  L = LEN(N$)
FOR I = 1 TO L
  N = VAL(MID$(N$, I, 1))
  FOR J = 2 TO 0 STEP -1
    X = INT(N / 2 ^ J): PRINT USING "#"; X;
    N = N - X * 2 ^ J
  NEXT J
NEXT I: PRINT
```

```
'2.3
' This program will add several items with tax (5%).
'
INPUT "Enter item: "; IT
WHILE IT <> -999
  T = T + IT
  INPUT "Enter item: "; IT
WEND
PRINT "SUBTOTAL = $"; T
TAX = INT((T * .05) * 100 + .5) / 100
PRINT "TAX      = $  "; TAX
PRINT "TOTAL    = $"; T + TAX
```

```
'2.4
' This program will divide the screen into 4 rectangles.
'
INPUT "Enter character: "; N$
CLS
FOR I = 1 TO 24
  IF I <> 12 THEN PRINT SPACE$(39); N$
  IF I = 12 THEN FOR J = 1 TO 79: PRINT N$; : NEXT J
NEXT I
```

```
'2.5
' This program will print the greatest and least in a set.
'
MAX = -900: MIN = 900
INPUT "Enter number: "; NUM
WHILE NUM <> -999
  IF NUM < MIN THEN MIN = NUM
  IF NUM > MAX THEN MAX = NUM
  INPUT "Enter number: "; NUM
WEND
PRINT "GREATEST = "; MAX
PRINT "LEAST = "; MIN
```

```
'2.6
' This program will print the sum, mean, median.
'
FOR I = 1 TO 10
  INPUT "Enter number: "; A(I):  S = S + A(I)
NEXT I
FOR I = 1 TO 9
  FOR J = I + 1 TO 10
    IF A(I) > A(J) THEN SWAP A(J), A(I)
  NEXT J
NEXT I
PRINT "SUM ="; S
PRINT "MEAN ="; S / 10
PRINT "MEDIAN ="; (A(5) + A(6)) / 2
```

```
'2.7
' This program will reverse the words in a sentence.
' Assume 1 space between each word.
'
INPUT "Enter sentence: "; A$: NUM = 1
FOR I = 1 TO LEN(A$)
  C$ = MID$(A$, I, 1)
  IF C$ <> " " THEN W$(NUM) = W$(NUM) + C$ ELSE NUM = NUM + 1
NEXT I
FOR I = NUM TO 1 STEP -1: PRINT W$(I); " "; : NEXT I
```

```
'2.8
' This program will convert cubic feet to cubic meters.
'
INPUT "Enter cubic feet: "; CF
C3 = CF * (12 * 2.54) ^ 3
CM = C3 / 100 / 100 / 100
PRINT USING "###.#### CUBIC METERS"; CM
```

```
'2.9
' This program will find sum of Ys and Xs for Y=2(X+5).
'
INPUT "Enter a, b: "; A, B
FOR X = A TO B: S = S + 2 * (X + 5): NEXT X
PRINT "SUM ="; S
```

```
'2.10
' This program will print 1 char. for 10 sec, 2 for 10 sec...
'
INPUT "Enter character: "; A$: CLS
FOR I = 1 TO 10
  FOR J = 1 TO I: PRINT A$; : NEXT J
  FOR J = 1 TO 2500 * 10: NEXT J
  CLS
NEXT I
```

```
'3.1
' This program converts a number from one base to another.
'
INPUT "NUMBER"; N$
INPUT "BASE"; B
INPUT "CONVERT TO BASE"; C
L = LEN(N$)
FOR I = 1 TO L
  X = VAL(MID$(N$, I, 1))
  S = S + X * B ^ (L - I)
NEXT I
J = INT(LOG(S) / LOG(C))
FOR I = J TO 0 STEP -1
  Y = INT(C ^ I + .1)
  X = INT(S / Y): NUM = NUM + X * 10 ^ I
  S = S - X * Y
NEXT I
PRINT NUM
```

```
'3.2
' This program will determine what triangle is made w/3 points.
'
INPUT "Enter X1, Y1: "; X1, Y1
INPUT "Enter X2, Y2: "; X2, Y2
INPUT "Enter X3, Y3: "; X3, Y3
D1 = SQR((X1 - X2) * (X1 - X2) + (Y1 - Y2) * (Y1 - Y2))
D2 = SQR((X2 - X3) * (X2 - X3) + (Y2 - Y3) * (Y2 - Y3))
D3 = SQR((X3 - X1) * (X3 - X1) + (Y3 - Y1) * (Y3 - Y1))
IF (D1 + D2 = D3) OR (D1 + D3 = D2) OR (D2 + D3 = D1) THEN
  PRINT "NOT A TRIANGLE": END
END IF
IF (D1 = D2) AND (D2 = D3) THEN PRINT "EQUILATERAL": END
IF (D1 = D2) OR (D2 = D3) OR (D1 = D3) THEN
  PRINT "ISOSCELES"
ELSE
  PRINT "SCALENE"
END IF
```

```
'3.3
' This program randomly selcts an X, Y in 10 x 10 grid.  User
' guesses numbers; if guess is wrong, a direction is given.
'
RANDOMIZE TIMER
X = INT(RND(4) * 9) + 1: Y = INT(RND(4) * 9) + 1
WHILE (A <> X) OR (B <> Y)
  INPUT "Enter X, Y:"; A, B
  IF A = X AND B < Y THEN PRINT "UP"
  IF A = X AND B > Y THEN PRINT "DOWN"
  IF A > X AND B = Y THEN PRINT "LEFT"
  IF A < X AND B = Y THEN PRINT "RIGHT"
  IF A < X AND B < Y THEN PRINT "UP AND RIGHT"
  IF A < X AND B > Y THEN PRINT "DOWN AND RIGHT"
  IF A > X AND B < Y THEN PRINT "UP AND LEFT"
  IF A > X AND B > Y THEN PRINT "DOWN AND LEFT"
WEND
```

```
'3.4
' This program will divide 1st number by 2nd out to N places.
'
INPUT "FIRST NUMBER"; N
INPUT "SECOND NUMBER"; D
INPUT "NUMBER OF DECIMAL PLACES"; P
X = INT(N / D):   PRINT USING "#."; X;
N = N - X * D
FOR I = 1 TO P
  N = N * 10: X = INT(N / D): PRINT USING "#"; X;
  N = N - X * D
NEXT I: PRINT
```

```
'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 store a list of words and provide options.
'
WHILE OP <= 3
  PRINT
  PRINT "1. ADD A WORD TO THE LIST"
  PRINT "2. DELETE A WORD FROM THE LIST"
  PRINT "3. DISPLAY THE ENTIRE LIST"
  INPUT OP
  IF OP = 1 THEN
    NUM = NUM + 1:  INPUT "Enter word: "; W$(NUM)
  ELSEIF OP = 2 THEN
    INPUT "Enter word: "; DEL$:  I = 1
    WHILE (I <= NUM) AND (W$(I) <> DEL$)
      I = I + 1
    WEND
    FOR J = I TO NUM - 1: W$(J) = W$(J + 1): NEXT J
    NUM = NUM - 1
  ELSEIF OP = 3 THEN
    FOR I = 1 TO NUM:  PRINT W$(I): NEXT I
  END IF
WEND
```

```
'3.7
' 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.
'
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$: J = J + 1: WEND
  FL(I) = J
  IF J = I THEN NL = NL + 1: UL(NL) = I  'A new letter
NEXT I
'
FOR N1 = 10 TO 98           'N1 must be 2 digits, >9
  FOR N2 = 100 - N1 TO 98   'N2 must be 2 digits, >9
    SUM = N1 + N2           'Sum must be 3 digits >99
    N1$ = MID$(STR$(N1), 2)
    N2$ = MID$(STR$(N2), 2)
    SUM$ = MID$(STR$(SUM), 2)
    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
    END IF
  NEXT N2
NEXT N1
IF TOT = 0 THEN PRINT "NO SOLUTION POSSIBLE"
```

```
'3.8
' This program will simulate random frog jumps on nine pads.
'
RANDOMIZE TIMER
CLS
FOR I = 1 TO 10
  LOCATE 1, 1: PRINT SPACE$(40);
  LOCATE 2, 1: PRINT "- - - - - - - - -"
  F = 5
  LOCATE 1, F * 2 - 1: PRINT "F": NUM = 0
  WHILE (F > 1) AND (F < 9)
    IF INT(RND(3) * 2) = 1 THEN F = F + 1 ELSE F = F - 1
    LOCATE 1, 1: PRINT SPACE$(40);
    LOCATE 1, F * 2 - 1: PRINT "F"
    FOR J = 1 TO 25: NEXT J
    NUM = NUM + 1
  WEND
  LOCATE 5, I * 3: PRINT NUM
NEXT I
```

```
'3.9
' This program will allow a user to position a cursor under a
' sentence using the L and R keys.  Space bar deletes letter.
'
CLS : INPUT "Enter sentence:"; S$: COL = 18
WHILE (CH$ <> CHR$(27)) AND (LEN(S$) > 1)
  LOCATE 2, COL, 1
  DO: CH$ = INKEY$: LOOP UNTIL CH$ > ""
  IF CH$ = "R" THEN COL = COL + 1
  IF CH$ = "L" THEN COL = COL - 1
  IF CH$ = " " THEN
    L = LEN(S$)
    S$ = LEFT$(S$, COL - 18) + RIGHT$(S$, L - (COL - 17))
  END IF
  LOCATE 1, 18: PRINT S$; " "
WEND
```

```
'3.10
' This program will simulate the movement of a pool ball on a
' rectangular pool table.  It moves in a 45 degree angle.
'
INPUT "Enter Width, Length: "; W, L
CLS : SCREEN 1
WI = 8: LI = 8
FOR I = 0 TO W: LINE (0, I * WI)-(L * LI, I * WI): NEXT I
FOR I = 0 TO L: LINE (I * LI, 0)-(I * LI, W * WI): NEXT I
X = 0: Y = W * WI: XD = 1: YD = -1
WHILE FI = 0
  PSET (X, Y), 0
  X = X + XD: Y = Y + YD
  PSET (X, Y), 1
  IF X = 0 OR X = L * LI THEN XD = -1 * XD
  IF Y = 0 OR Y = W * WI THEN YD = -1 * YD
  FINISHED = 1: LOCATE 20, 1
  IF X = 0 AND Y = 0 THEN PRINT "LEFT-TOP":           FI = 1
  IF X = 0 AND Y = W * WI THEN PRINT "LEFT-BOTTOM":   FI = 1
  IF X = L * LI AND Y = 0 THEN PRINT "RIGHT-TOP":     FI = 1
  IF X = L * LI AND Y = W * WI THEN PRINT "RIGHT-BOTTOM": FI = 1
WEND
DO: A$ = INKEY$: LOOP UNTIL A$ <> ""
SCREEN 0: WIDTH 80: CLS
```

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '84
BASIC PROGRAM SOLUTIONS

```
'1.1
' This program will produce a table of Fahrenheit for Celcuis.
'
PRINT "CELCIUS", "FAHRENHEIT"
FOR C = 50 TO 200 STEP 25
  PRINT C, INT(1.8 * C + 32 + .5)
NEXT C
```

```
'1.2
' This program will determine time a person slept in seconds.
'
INPUT "WHAT TIME DID YOU GO TO BED (H, M, S)"; H1, M1, S1
INPUT "WHAT TIME DID YOU GET UP (H, M, S)"; H2, M2, S2
T = (11 - H1) * 3600 + (59 - M1) * 60 + (60 - S1)
PRINT "YOU SLEPT FOR"; T + H2 * 3600 + M2 * 60 + S2; "SECONDS"
```

```
'1.3
'This program will display distance/height of a golf ball.
'
PRINT "T", "H", "V"
WHILE (V > 0) OR (T < 1)
  H = 120 * T:  V = 120 * T - 16 * T * T
  PRINT T, H, V
  T = T + .5
WEND
```

```
'1.4
' This program will produce table of mice population and food.
'
PRINT "NUMBER OF YEARS", "POPULATION", "FOOD SUPPLY FOR"
Y = 0: P = 10: F = 100
PRINT Y, , P, F
WHILE P < F
  Y = Y + 1: P = P * 2:  F = F + 40
  PRINT Y, , P, F
WEND
```

```
'1.5
' This program will determine time that a savings doubles.
'
INPUT "Enter amount, %"; N, P
X = N: Y = 0
WHILE X < 2 * N
  X = X * (1 + P / 100)
  Y = Y + 1
WEND
PRINT Y; "YEARS"
```

```
'1.6
' This program will determine name at beginning and end.
'
MIN$ = "ZZZZZ": MAX$ = "AAAAAA"
FOR I = 1 TO 5
  INPUT NM$
  IF NM$ < MIN$ THEN MIN$ = NM$
  IF NM$ > MAX$ THEN MAX$ = NM$
NEXT I
PRINT "NAME CLOSEST TO BEGINNING: "; MIN$
PRINT "NAME CLOSEST TO END: "; MAX$
```

```
'1.7
' This program will determine longest run of heads of tosses.
'
RANDOMIZE TIMER
INPUT "N: "; N: H = 0: MAX = 0
FOR I = 1 TO N
  X = RND(2): IF X < .5 THEN H = H + 1
  IF X >= .5 THEN IF H > MAX THEN MAX = H: H = 0 ELSE H = 0
NEXT I
IF H > MAX THEN MAX = H
PRINT MAX; "CONSECUTIVE HEADS"
```

```
'1.8
' This program will display numbers with 7s zapped.
'
FOR I = 1 TO 100
 T = INT(I / 10): O = I - T * 10: D7 = (I / 7 = INT(I / 7))
 IF (T = 7 OR O = 7) AND D7 THEN
   PRINT "ZAPZAP",
 ELSEIF (T = 7 OR O = 7) THEN
   PRINT "ZAP",
 ELSE
   PRINT I,
 END IF
NEXT I
```

```
'1.9
' This program will print the # of double letters.
'
INPUT "Enter text: "; A$
FOR I = 1 TO LEN(A$)
  C$ = MID$(A$, I, 1)
  IF C$ = LC$ THEN D = D + 1
  LC$ = C$
NEXT I
PRINT D
```

```
'1.10
' This program will display sevens multiplication facts.
'
FOR I = 0 TO 9
  W = 0
  DO
    PRINT I; "X 7 = "; : INPUT ANS
    IF ANS <> I * 7 THEN
      IF W = 0 THEN
        W = 1
      ELSE
        PRINT I * 7: W = 2
      END IF
    END IF
  LOOP UNTIL (I * 7 = ANS) OR (W = 2)
NEXT I
```

```
'2.1
' This program will print number of vowels in text.
'
INPUT "Enter text: "; A$
FOR I = 1 TO LEN(A$)
  C$ = MID$(A$, I, 1)
  IF C$ = "A" OR C$ = "E" OR C$ = "I" OR C$ = "O" OR C$ = "U" THEN
    V = V + 1
  END IF
NEXT I
PRINT V; "VOWELS"
```

```
'2.2
' This program sorts rational numbers in increasing order.
'
INPUT "Enter N, M: "; N, M
WHILE N > 0 AND M > 0
  S = S + 1: A$(S) = STR$(N) + "/" + STR$(M): V(S) = N / M
  INPUT "Enter N, M: "; N, M
WEND
FOR I = 1 TO S - 1
  FOR J = I + 1 TO S
    IF V(I) > V(J) THEN SWAP V(I), V(J): SWAP A$(I), A$(J)
  NEXT J
NEXT I
FOR I = 1 TO S: PRINT A$(I): NEXT I
```

```
'2.3
' This program displays #s that sum of cubes of digits= #.
'
FOR I = 1 TO 9
  FOR J = 0 TO 9
    FOR K = 0 TO 9
      NUM = I * 100 + J * 10 + K
      IF NUM = I * I * I + J * J * J + K * K * K THEN PRINT NUM
    NEXT K
  NEXT J
NEXT I
```

```
'2.4
' This program will print a triangle of #s by an algorithm.
'
INPUT "Enter # of rows: "; N
FOR I = 1 TO N
  PRINT SPACE$(N - I);
  FOR J = I TO 2 * I - 1
    X = J - INT(J / 10) * 10
    PRINT MID$(STR$(X), 2);
  NEXT J
  IF I > 1 THEN
    FOR J = 2 * I - 2 TO I STEP -1
      X = J - INT(J / 10) * 10
      PRINT MID$(STR$(X), 2);
    NEXT J
  END IF
  PRINT
NEXT I
```

```
'2.5
' This program will display a page of multipilcation drills.
'
RANDOMIZE TIMER: CLS
PRINT "     MULTIPLICATION DRILL"
FOR I = 1 TO 6
  H = INT((I - 1) / 3): V = I - H * 3: H = H * 20 + 1
  X = INT(RND(3) * 90 + 10):  Y = INT(RND(3) * 9 + 1)
  LOCATE V * 5, H: PRINT MID$(STR$(I), 2); ". "; X
  LOCATE V * 5 + 1, H: PRINT "   X"; Y
  LOCATE V * 5 + 2, H: PRINT "   ----"
NEXT I
```

```
'2.6
' This program will simulate throwing darts.
'
RANDOMIZE TIMER: INPUT "Enter N: "; N
FOR I = 1 TO N
  X = INT(RND(3) * 5):  Y = INT(RND(3) * 5)
  A(X, Y) = 1
NEXT I
FOR I = 0 TO 4
  FOR J = 0 TO 4
    IF A(I, J) THEN PRINT "* "; : S = S + 1 ELSE PRINT ". ";
  NEXT J: PRINT
NEXT I
PRINT "NUMBER OF THROWS ="; N
PRINT "NUMBER OF HITS ="; S
```

```
'2.7
' This program will determine if text is palindrome.
'
PRINT "Enter text: "; : A$ = ""
WHILE A$ <> CHR$(13)
  DO: A$ = INKEY$: LOOP UNTIL A$ <> ""
  PRINT A$; : IF A$ >= "A" AND A$ <= "Z" THEN S$ = S$ + A$
WEND
L = LEN(S$)
FOR I = 1 TO L / 2
  IF MID$(S$, I, 1) <> MID$(S$, L - I + 1, 1) THEN
    PRINT "NOT A PALINDROME": END
  END IF
NEXT I
PRINT "A PALINDROME"
```

```
'2.8
' This program will display the frequency of letters.
'
DIM A(26): INPUT "Enter sentence:"; A$: L = LEN(A$)
FOR I = 1 TO L
  C$ = MID$(A$, I, 1)
  IF C$ >= "A" AND C$ <= "Z" THEN
    X = ASC(C$) - ASC("A") + 1: A(X) = A(X) + 1: T = T + 1
  END IF
NEXT I
PRINT "LETTER", "FREQUENCY", "PERCENT"
FOR I = 1 TO 26
  IF A(I) > 0 THEN
    PRINT CHR$(I + 64), A(I), INT(A(I) / T * 100 + .5)
  END IF
NEXT I
PRINT "TOTAL"; T
```

```
'2.9
' This program will print longest word in sentence.
'
INPUT "Enter sentence:"; A$: A$ = A$ + " ": L = LEN(A$)
FOR I = 1 TO L
  C$ = MID$(A$, I, 1)
  IF C$ <> " " THEN
    W$ = W$ + C$
  ELSE
    IF LEN(W$) > LEN(MAX$) THEN MAX$ = W$
    W$ = ""
  END IF
NEXT I
PRINT MAX$
```

```
'2.10
' This program will play rock, scissors, and paper.
'
RANDOMIZE TIMER
INPUT "Enter R, S, P, or Q: "; A$
WHILE A$ <> "Q"
  X = INT(RND(3) * 3)
  IF X = 0 AND A$ = "R" THEN
    T = T + 1: PRINT "TIE"
  ELSEIF X = 1 AND A$ = "S" THEN T = T + 1: PRINT "TIE"
  ELSEIF X = 2 AND A$ = "P" THEN T = T + 1: PRINT "TIE"
  ELSEIF X = 0 AND A$ = "P" THEN W = W + 1: PRINT "YOU WIN"
  ELSEIF X = 1 AND A$ = "R" THEN W = W + 1: PRINT "YOU WIN"
  ELSEIF X = 2 AND A$ = "S" THEN W = W + 1: PRINT "YOU WIN"
  ELSE
    L = L + 1: PRINT "I WIN"
  END IF
  INPUT "Enter R, S, P, or Q: "; A$
WEND
PRINT T; "TIES"
PRINT W; "WINS (YOURS)"
PRINT L; "LOSSES (MINE)"
```

```
'3.1
' This program will display a random trail of asterisks.
'   The program description is poorly worded and ambiguous.
'   The judging criteria is also described poorly.
'
RANDOMIZE TIMER
DIM A(24, 80)
DO
  CLS : FOR I = 1 TO 24: FOR J = 1 TO 80: A(I, J) = 0: NEXT J, I
  V = 12: H = 40: A(V, H) = 1
  LOCATE V, H: PRINT "S": SAMERUN = -1
  WHILE SAMERUN
    DO
      X = INT(RND(3) * 4)
    LOOP UNTIL (X - 2 <> Y) AND (Y - 2 <> X)
    H = H + (X = 0) - (X = 2)
    V = V + (X = 1) - (X = 3)
    IF A(V, H) OR V = 0 OR V = 23 OR H = 0 OR H = 80 THEN
      LOCATE 22, 1: PRINT "THE MAXIMUM DISTANCE FROM START = ";
      PRINT ABS(40 - H) + ABS(12 - V)
      DO: A$ = INKEY$: LOOP UNTIL A$ <> ""
      SAMERUN = 0
    ELSE
      A(V, H) = 1: LOCATE V, H: PRINT "*": Y = X
    END IF
  WEND
LOOP UNTIL A$ = "Q"
```

```
'3.2
' This program will decode a message with frequent letters.
'
DIM A$(32), A(32), B$(32)
B$ = "ETAOINSHRDLU"
INPUT "Message: "; A$: L = LEN(A$)
FOR I = 1 TO L: A$(I) = MID$(A$, I, 1): NEXT I
FOR I = 1 TO L
  K = 0
  WHILE (A$(K) <> A$(I)) AND (K <= I - 1)
    K = K + 1
  WEND
'   Found 1st occurence of a letter, count occurences after it
  IF K = I THEN
    FOR J = I TO L
      IF A$(I) = A$(J) THEN A(I) = A(I) + 1
    NEXT J
    IF A(I) > G THEN G = A(I)
  END IF
NEXT I
' Replace letters in message
FOR I = G TO 1 STEP -1
  J = 1
  WHILE (A(J) <> I) AND (J <= L)
    J = J + 1
  WEND
  IF J <= L THEN
    S = S + 1
    FOR K = J TO L
      IF A$(K) = A$(J) THEN B$(K) = MID$(B$, S, 1)
    NEXT K
  END IF
NEXT I
FOR I = 1 TO L: PRINT B$(I); : NEXT I
```

```
'3.3
' This program will produce the digital product root.
'
INPUT "ORIGINAL VALUE (1 TO 7 DIGITS): "; N
PRINT N
WHILE N > 9
  A$ = MID$(STR$(N), 2):  L = LEN(A$): N = 1
  FOR I = 1 TO L
    X = VAL(MID$(A$, I, 1))
    IF X > 0 THEN N = N * X
  NEXT I
  PRINT N
WEND
```

```
'3.4
' This program will display twin primes.
'
INPUT "Enter N: "; N
PRINT "TWIN PRIMES NOT GREATER THAN"; N
FOR I = 3 TO N - 2
  FOR J = 2 TO SQR(I)
    IF I MOD J = 0 THEN GOTO NextPrime
  NEXT J
  T = I + 2
  FOR J = 2 TO SQR(T)
    IF T MOD J = 0 THEN GOTO NextPrime
  NEXT J: PRINT I, T
NextPrime: NEXT I
```

```
'3.5
' This program will print subsets of m people.
'
DIM A(26), A$(26): INPUT "INPUT NUMBER, CAPACITY"; L, M
FOR I = 1 TO M: A(I) = M - I + 1:     NEXT I
FOR I = 1 TO L: A$(I) = CHR$(64 + I): NEXT I
N = 1:  A(1) = A(1) - 1
WHILE N <= M
  A(N) = A(N) + 1
  IF N > 1 THEN
    FOR I = N - 1 TO 1 STEP -1: A(I) = A(I + 1) + 1: NEXT I
  END IF
  IF A(N) <= L - N + 1 THEN
    FOR I = M TO 1 STEP -1: PRINT A$(A(I)); : NEXT I: PRINT ,
    S = S + 1: N = 0
  END IF
  N = N + 1
WEND
PRINT : PRINT "THERE ARE"; S; "SUBSETS"
```

```
'3.6
' This program will display histogram of letter frequency.
'
DIM A(256): CLS
DATA "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG."
DATA "THIS IS AN EXAMPLE OF HOW"
DATA "TO TEST YOUR HISTOGRAM PROGRAM.  YOU"
DATA "CAN USE THIS EXAMPLE."
DATA "*END*"
READ A$
WHILE A$ <> "*END*"
  FOR I = 1 TO LEN(A$)
    X = ASC(MID$(A$, I, 1))
    IF X >= 65 AND X <= 90 THEN A(X) = A(X) + 1
    IF A(X) > G THEN G = A(X)
  NEXT I
  READ A$
WEND
FOR I = G TO 1 STEP -1
  FOR J = 65 TO 90
    IF A(J) >= I THEN
      LOCATE G - I + 1, J - 64: PRINT "*";
    END IF
  NEXT J: PRINT
NEXT I
FOR I = 65 TO 90: PRINT CHR$(I); : NEXT I
```

```
'3.7
' This program will display a repeating decimal.
'
DIM R(100)
INPUT "Enter N, D: "; N, D
PRINT N; "/"; D; "= "; : X = INT(N / D)
IF X > 0 THEN PRINT MID$(STR$(X), 2);
PRINT ".";
DO
  I = I + 1: R = N - D * X
  IF R = 0 THEN PRINT A$: END
  R(I) = R:  N = R * 10: X = INT(N / D)
'   Display decimal if remainder repeats itself
  FOR J = 1 TO I - 1
    IF R(J) = R THEN
      PRINT LEFT$(A$, J - 1); "("; RIGHT$(A$, I - J); ")": END
    END IF
  NEXT J
  A$ = A$ + MID$(STR$(X), 2)
LOOP
```

```
'3.8
' This program will print # of round numbers less than N.
'
INPUT "INPUT NUMBER"; N
FOR I = 2 TO N
  M = I:  S = 0:  K = INT(LOG(M) / LOG(2))
  FOR J = K TO 0 STEP -1
    X = INT(M / 2 ^ J)
    S = S + X:  M = M - X * 2 ^ J
  NEXT J
  IF S + S = K + 1 THEN T = T + 1
NEXT I
PRINT "THERE ARE"; T; "ROUND NUMBERS LESS THAN OR EQUAL TO"; N
```

```
'3.9
' This program will provide automated price increases.
'
INPUT "Enter %"; P: P = P / 100
READ A$
WHILE A$ <> "*END*"
  L = LEN(A$): I = 0
  DO
    WHILE (I < L) AND (X$ <> "$")
       I = I + 1: X$ = MID$(A$, I, 1): PRINT X$;
    WEND
    IF X$ <> "$" THEN
      PRINT
    ELSE
      J = I:  X = 50
'
  WHILE (J < L) AND (X$ = "." OR (X > 47 AND X < 58)) AND PER < 2
    J = J + 1: X$ = MID$(A$, J, 1): X = ASC(X$)
    IF X$ = "." THEN PER = PER + 1
  WEND
'
      T = VAL(MID$(A$, I + 1, J - I))
      T = T + T * P: T = INT(T * 100 + .5) / 100
      PRINT MID$(STR$(T), 2);
      I = J - 1: PER = 0
    END IF
  LOOP UNTIL I >= L
  READ A$
WEND
DATA "THE CURRENT COST OF BUCKLES IS"
DATA "3 FOR $2.50, OR $10.00 A DOZEN."
DATA "*END*"
```

```
'3.10
' This program will simulate tennis sets between 2 players.
'
RANDOMIZE TIMER
INPUT "NUMBER OF SETS ="; N
INPUT "% CHANCE A WINS A POINT="; P
DO
  IF RND(2) < P / 100 THEN A = A + 1 ELSE B = B + 1
  IF A > 3 AND A > B + 1 THEN
    PRINT "A"; : AG = AG + 1: A = 0: B = 0
  END IF
  IF B > 3 AND B > A + 1 THEN
    PRINT "B"; : BG = BG + 1: A = 0: B = 0
  END IF
  IF AG > 5 AND AG > BG + 1 THEN
    PRINT " (A)": W = W + 1: AG = 0: BG = 0
  END IF
  IF BG > 5 AND BG > AG + 1 THEN
    PRINT " (B)": L = L + 1: AG = 0: BG = 0
  END IF
LOOP UNTIL W + L = N
IF W > L THEN PRINT "PLAYER 'A' WON"; W; "SETS OUT OF"; N: END
PRINT "PLAYER 'B' WON"; L; "SETS OUT OF"; N
```

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '80 }
{ -- PASCAL PROGRAM SOLUTIONS }


{1.1}
program One1T80;
{ -- This program will print terms of the Fibinacci sequence. }
  var
    N, I: Integer;
    A:    Array[1..99] of LongInt;

begin
  Write ('Enter number of terms: ');  Readln (N);
  A[1] := 1;  A[2] := 1;
  for I := 3 to N do
    A[I] := A[I-1] + A[I-2];
  for I := 1 to N do  Write (A[I], '  ');
  Writeln;
end.


{1.2}
program One2T80;
{ -- This program will flash a symbol on/off every 10 seconds. }
uses Crt;
  var
    I: Byte;

begin
  ClrScr;
  for I := 1 to 5 do begin
    Write ('*');
    delay (500);
    ClrScr;
    delay (6000);
  end;
end.


{1.3}
program One3T80;
{ -- This program will display the perimeter of a rectangle. }
  var
    L, W: Integer;

begin
  Write ('Enter L, W: ');  Readln (L, W);
  Writeln (L + L + W + W);
end.
```

```
{1.4}
program One4T80;
{ -- This program will convert Celcius to Fahrenheit. }
  var
    C: Integer;

begin
  Write ('Enter C: ');  Readln (C);
  Writeln ( (C * 9) / 5 + 32 :2:0);
end.
```

```
{1.5}
program One5T80;
{ -- This program will determine if input is numeral or not. }
  var
    Ch: Char;

begin
  Write ('Enter character: ');  Readln (Ch);
  if Ch in ['0' .. '9'] then
    Writeln ('NUMERAL')
  else
    Writeln ('NON-NUMERAL');
end.
```

```
{1.6}
program One6T80;
{ -- This program will determine gas mileage. }
  var
    D, G: Integer;

begin
  Write ('Enter D, G: ');  Readln (D, G);
  Writeln (D div G);
end.
```

```
{1.7}
program One7T80;
{ -- This program will test if a number is divisible by 5. }
  var
    N: Integer;

begin
  Write ('Enter number: ');  Readln (N);
  if N mod 5 = 0 then
    Writeln ('YES')
  else
    Writeln ('NO');
end.
```

```
{1.8}
program One8T80;
{ -- This program will print the length of a side of a triangle. }
  var
    L, H: Integer;

begin
  Write ('Enter L, H: ');  Readln (L, H);
  Writeln (Sqrt(H*H - L*L) :1:0);
end.
```

```
{1.9}
program One9T80;
{ -- This program will move a blob across the screen. }
uses Crt;
  var
    C: Integer;

begin
  ClrScr;
  for C := 1 to 79 do begin
    GotoXY (C,2);  Write ('*');
    Delay (10);
    GotoXY (C,2);  Write (' ');
  end;
end.
```

```
{2.1}
program Two1T80;
{ -- This program will print the largest number in a sequence. }
  var
    N, I, X, Max: Integer;

begin
  Write ('Enter number of #s: ');  Readln (N);
  Max := -999;
  for I := 1 to N do begin
    Write ('Enter #: ');  Readln (X);
    if X > Max then Max := X;
  end;
  Writeln (Max);
end.


{2.2}
program Two2T80;
{ -- This program will determine what figure is made. }
  var
    S1, S2, S3, S4: Integer;

begin
  Write ('Enter 4 sides: ');  Readln (S1, S2, S3, S4);
  if S4 = 0 then
    Writeln ('TRIANGLE')
  else if (S1 = S2) and (S2 = S3) and (S3 = S4) then
    Writeln ('SQUARE')
  else
    Writeln ('RECTANGLE');
end.


{2.3}
program Two3T80;
{ -- This program will sum numbers from 1000 to 2000. }
  var
    I: Integer;
    S: LongInt;

begin
  S := 0;
  for I := 1000 to 2000 do  S := S + I;
  Writeln (S);
end.
```

```
{2.4}
program Two4T80;
{ -- This program will reverse a 3 digit number. }
  var
    N: String[3];
    I: Byte;

begin
  Write ('Enter number: ');  Readln (N);
  for I := 3 downto 1 do
    Write (Copy(N, I, 1));
  Writeln;
end.
```

```
{2.5}
program Two5T80;
{ -- This program will draw a rectangle on the screen. }
uses Crt;
  var
    I: Byte;

begin
  ClrScr;
  for I := 1 to 20 do Write ('*');  Writeln;
  for I := 1 to 5 do Writeln ('*', ' ':18, '*');
  for I := 1 to 20 do Write ('*');
end.
```

```
{2.6}
program Two6T80;
{ -- This program will print 3 numbers in increasing order. }
  var
    I, J, X: Integer;
    A:       Array [1..3] of Integer;

begin
  Write ('Enter 3 number: ');  Readln (A[1], A[2], A[3]);
  for I := 1 to 2 do
    for J := I+1 to 3 do
      if A[I] > A[J] then begin
        X := A[I];  A[I] := A[J];  A[J] := X;
      end;
  for I := 1 to 3 do Write (A[I], '  ');
  Writeln;
end.
```

```
{2.7}
program Two7T80;
{ -- This program will determine mean of set of numbers. }
  var
    N, I, X, S: Integer;

begin
  Write ('Enter number of #s: ');  Readln (N);
  S := 0;
  for I := 1 to N do begin
    Write ('Enter #: ');  Readln (X);
    S := S + X;
  end;
  Writeln (S / N :3:1);
end.
```

```
{2.8}
program Two8T80;
{ -- This program will determine if a number is even or odd. }
  var
    N: Integer;

begin
  Write ('Enter number: ');  Readln (N);
  if N mod 2 = 0 then
    Writeln ('EVEN')
  else
    Writeln ('ODD');
end.
```

```
{2.9}
program Two9T80;
{ -- This program will determine if a number is prime. }
  var
    N, I, Sq: Integer;

begin
  Write ('Enter number: ');  Readln (N);
  Sq := Trunc( Sqrt(N) );
  if N mod 2 = 0 then begin
     Writeln ('NO');  Exit; end
  else
    for I := 3 to Sq do
      if N mod I = 0 then begin
        Writeln ('NO');  Exit;  end;
  Writeln ('YES');
end.
```

```
{2.10}
program Two10T80;
{ -- This program will compute value of change. }
  const
    Coin: Array [1..4] of String[8] =
         ('QUARTERS', 'DIMES', 'NICKELS', 'PENNIES');
    Amount: Array[1..4] of Integer = (25, 10, 5, 1);
  var
    I, X, S: Integer;

begin
  S := 0;
  for I := 1 to 4 do begin
    Write ('How many ', Coin[I], '? ');  Readln (X);
    S := S + Amount[I] * X;
  end;
  Writeln (S, ' CENTS');
end.
```

```
{2.11}
program Two11T80;
{ -- This program will count number of e's in sentence. }
  var
    I, E: Byte;
    Sent: String[80];

begin
  Write ('Enter sentence: ');  Readln (Sent);
  E := 0;
  for I := 1 to Length(Sent) do
    if Copy(Sent, I, 1) = 'E' then Inc(E);
  Writeln (E);
end.
```

```
{3.1}
program Thr1T80;
{ -- This program allows user to answer multiplication facts. }
  var
    X, Y, N: Integer;

begin
  Randomize;  X := Random(13);  Y := Random(13);
  Write (X, ' X ', Y, ' = ');  Readln (N);
  if X * Y = N then
    Writeln ('RIGHT')
  else
    Writeln ('WRONG');
end.
```

```
{3.2}
program Thr2T80;
{ -- This program will randomize the digits 0..9. }
  var
    I, X, Y: Byte;
    A: Array [0..9] of Byte;

begin
  Randomize;
  for I := 0 to 9 do A[I] := I;
  for I := 0 to 9 do begin
    X := Random(10);
    Y := A[I];  A[I] := A[X];  A[X] := Y;
  end;
  for I := 0 to 9 do Write (A[I], ' ');
  Writeln;
end.
```

```
{3.3}
program Thr3T80;
{ -- This program will round a number to nearest ten. }
  var
    N: Integer;

begin
  Write ('Enter number: ');  Readln (N);
  Writeln (Trunc((N+5)/10) * 10);
end.
```

```
{3.4}
program Thr4T80;
{ -- This program will change a number from base 10 to 4. }
  var
    N, I, J, X, Pow: Integer;

begin
  Write ('Enter number: ');  Readln (N);
  J := Trunc(Ln(N) / Ln(4));
  Pow := 1;
  for I := 0 to J do Pow := Pow * 4;
  for I := J downto 0 do begin
    Pow := Pow div 4;
    X := Trunc(N / Pow);  Write (X);
    N := N - X * Pow;
  end;
  Writeln;
end.
```

```
{3.5}
program Thr5T80;
{ -- This program will change a number from base 3 to 10. }
  var
    N: String[7];
    I, L, S, X, Code, Pow: Integer;

begin
  Write ('Enter number: ');  Readln (N);
  L := Length(N);  Pow := 1;  S := 0;
  for I := L downto 1 do begin
    Val(Copy(N, I, 1), X, Code);
    S := S + X * Pow;
    Pow := Pow * 3;
  end;
  Writeln (S);
end.
```

```
{3.6}
program Thr6T80;
{ -- This program will change a decimal to a fraction. }
  var
    A:                 Char;
    Num:               String[5];
    I, L, N, D, Code: Integer;
```

```
begin
  Write ('Enter decimal: ');  Readln (Num);
  Write ('R or T: ');  Readln (A);
  L := Length(Num) - 1;
  Val (Copy(Num, 2, L), N, Code);
  D := 1;
  for I := 1 to L do D := D * 10;
  if A = 'R' then D := D - D div 10;
  I := N;
  while (N mod I <> 0) or (D mod I <> 0) do
    Dec(I);
  Writeln (N div I, '/', D div I);
end.
```

```
{3.7}
program Thr7T80;
{ -- This program will represent and amount of money. }
  const
    Coin: Array [1..5] of String[8] =
          ('HALF', 'QUARTER', 'DIMES', 'NICKELS', 'PENNIES');
    Amount: Array [1..5] of Integer = (50, 25, 10, 5, 1);
  var
    C, X, I: Integer;

begin
  Write ('Enter cents: ');  Readln (C);
  for I := 1 to 5 do begin
    X := C div Amount[I];
    Writeln (X, ' ', Coin[I]);
    C := C - X * Amount[I];
  end;
end.
```

```
{3.8}
program Thr8T80;
{ -- This program will allow user to guess a generated #. }
  var
    X, G: Byte;

begin
  Randomize;
  X := Random(10) + 1;  G := 0;
  while X <> G do begin
    Write ('Enter guess: ');  Readln (G);
    if G > X then Writeln ('TOO HIGH');
    if G < X then Writeln ('TOO LOW');
  end;
  Writeln ('RIGHT ON');
end.
```

```
{3.9}
program Thr9T80;
{ -- This program will find values for a,b,c,d. }
  var
    A, B, C, D, N, M, I, APow, CPow: Integer;

begin
  for A := 1 to 9 do
    for B := 0 to 9 do
      for C := 0 to 9 do
        for D := 0 to 9 do begin
          N := A*1000 + B*100 + C*10 + D;
          APow := 1;
          for I := 1 to B do APow := APow * A;
          CPow := 1;
          for I := 1 to D do CPow := CPow * C;
          M := APow * CPow;
          if N = M then begin
            Writeln ('A=', A, '  B=', B, '  C=', C, '  D=', D);
            exit;
          end;
        end;
end.
```

```
{3.10}
program Thr10T80;
{ -- This program will print day of week for a date. }
  const
    Days: Array[1..12] of Integer =
          (31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
  var
    M, D, I, S, X: Integer;
    Name:          String[21];

begin
  Name := 'MONTUEWEDTHUFRISATSUN';
  Write ('Enter month, day: ');  Readln (M, D);  S := 0;
  for I := 1 to M-1 do
    S := S + Days[I];
  S := S + D;
  X := S mod 7;
  Writeln (Copy(Name, X*3+1, 3));
end.
```

```
{3.11}
program Thr11T80;
{ -- This program will simulate an "etch-a-sketch". }
uses Crt;
  var
    Row, Col: Byte;
    A:        Char;

begin
  ClrScr;  Row := 12;  Col := 40;
  repeat
    GotoXY (Col, Row);  Write ('*');
    A := Readkey;
    case A of
      'I':  Dec(Row);
      'M':  Inc(Row);
      'J':  Dec(Col);
      'K':  Inc(Col);
    end;
  until A = Char(27);
end.
```

```
{3.12}
program Thr12T80;
{ -- This program will determine if a word is a palindrome. }
  var
    A:      String[12];
    L, R:   String[1];
    I, Len: Byte;

begin
  Write ('Enter word: ');  Readln (A);
  Len := Length(A);  I := 1;
  repeat
    L := Copy (A, I, 1);
    R := Copy (A, Len-I+1, 1);
    if L <> R then Write ('NOT ');
    Inc(I);
  until (I = Len) or (L <> R);
  Writeln ('PALINDROME');
end.
```

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '81 }
{ -- PASCAL PROGRAM SOLUTIONS }


{1.1}
program One1T81;
{ -- This program will compute percent of heads and tails. }
  var
    H, T, S: Integer;

begin
  Write ('Enter number of heads: ');  Readln (H);
  Write ('Enter number of tails: ');  Readln (T);
  S := T + H;
  Writeln ('PERCENT HEADS: ', H/S * 100 :3:1);
  Writeln ('PERCENT TAILS: ', T/S * 100 :3:1);
end.


{1.2}
program One2T81;
{ -- This program will display the angle of a polygon. }
  var
    N: Integer;

begin
  Write ('Enter number of sides: ');  Readln (N);
  Writeln ('ANGLE=', 180 * (N - 2) div N);
end.


{1.3}
program One3T81;
{ -- This program will compute the value of a function. }
  var
    A, B, C, X: Integer;

begin
  Write ('A, B, C, X: ');  Readln (A, B, C, X);
  Writeln ('AX^2 + BX + C = ', A*X*X + B*X + C);
end.
```

```
{1.4}
program One4T81;
{ -- This program will compute the net price after discounts. }
  var
    I:    Byte;
    P, D: Real;

begin
  Write ('Enter original price: $');  Readln (P);
  for I := 1 to 2 do begin
    Write ('Enter discount ', I, ' percent: ');  Readln (D);
    P := P - P * D / 100
  end;
  Writeln ('FINAL NET PRICE: $', P :3:2);
end.
```

```
{1.5}
program One5T81;
{ -- This program will determine the quadrant of a point. }
  var
    X, Y: Integer;

begin
  Write ('Enter X, Y: ');  Readln (X, Y);
  if (X > 0) and (Y > 0) then Writeln ('QUADRANT: I');
  if (X < 0) and (Y > 0) then Writeln ('QUADRANT: II');
  if (X < 0) and (Y < 0) then Writeln ('QUADRANT: III');
  if (X > 0) and (Y < 0) then Writeln ('QUADRANT: IV');
  if X = 0 then Writeln ('LIES ON THE Y-AXIS');
  if Y = 0 then Writeln ('LIES ON THE X-AXIS');
end.
```

```
{2.1}
program Two1T81;
{ -- This program will sum two fractions. }
  var
    A, B, C, D, Num, Den, I: Integer;

begin
  Write ('Enter a, b, c, d: ');  Readln (A, B, C, D);
  Num := A * D + B * C;
  Den := B * D;
    I := Num;
  while (Num mod I <> 0) or (Den mod I <> 0) do
    Dec(I);
  Writeln (Num div I, '/', Den div I);
end.
```

```
{2.2}
program Two2T81;
{ -- This program will determine if quad is equilateral. }
  var
    I, Asq, Bsq: Integer;
    A, B:        Array [1..5] of Integer;
    C:           Array [1..5] of Real;

begin
  for I := 1 to 4 do begin
    Write ('Enter point ', I, ': ');  Readln (A[I], B[I]);
  end;
  A[5] := A[1];  B[5] := B[1];
  for I := 1 to 4 do begin
    Asq := (A[I] - A[I+1]) * (A[I] - A[I+1]);
    Bsq := (B[I] - B[I+1]) * (B[I] - B[I+1]);
    C[I] := Sqrt (Asq + Bsq);
  end;
  Write ('QUAD IS ');
  for I := 1 to 3 do
    if Abs (C[I] - C[I+1]) > 0.1 then begin
      Writeln ('NOT EQUILATERAL');  Exit;
    end;
  Writeln ('EQUILATERAL');
end.
```

```
{2.3}
program Two3T81;
{ -- This program will print discount rate for phone call. }
  var
    D, T: Integer;

begin
  Write ('Enter day, time: ');  Readln (D, T);
  if (T >= 1700) and (T < 2300) then Writeln ('20%') else
  if (T >= 2300) or  (T < 700)  then Writeln ('40%') else
  if D = 7 then Writeln ('20%') else
  if D = 1 then Writeln ('40%') else
  Writeln ('NO DISCOUNT');
end.
```

```
{2.4}
program Two4T81;
{ -- This program will determine if graph is parallel. }
  var
    A, B, C, D, E, F: Integer;

begin
  Write ('Enter A, B, C: ');  Readln (A, B, C);
  Write ('Enter D, E, F: ');  Readln (D, E, F);
  Write ('LINES ARE ');
  if A * E <> D * B then Write ('NOT ');
  Writeln ('PARALLEL');
end.
```

```
{2.5}
program Two5T81;
{ -- This program will find the LCM of 3 integers. }
  var
    A, B, C, S: Integer;

begin
  Write ('Enter three integers: ');
  Readln (A, B, C);  S := 0;
  repeat
    S := S + A;
  until (s mod B = 0) and (S mod C = 0);
  Writeln (S);
end.
```

```
{3.1}
program Thr1T81;
{ -- This program will convert a number from base 10 to B. }
  var
    N, B, J, I, X, Pow: Integer;

begin
  Write ('Enter numeral, base: ');  Readln (N, B);
  J := Trunc (Ln(N) / Ln(B));  Pow := 1;
  for I := 0 to J do  Pow := Pow * B;
  for I := J downto 0 do begin
    Pow := Pow div B;
    X := N div Pow;  Write (X);
    N := N - X * Pow;
  end;
  Writeln;
end.


{3.2}
program Thr2T81;
{ -- This program will print the mode in a list. }
  var
    N, I, J, Max: Integer;
    A, B:         Array [1..20] of Integer;

begin
  Write ('Enter how many numbers: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter #: ');  Readln (A[I]);
  end;
  for I := 1 to N do begin
    B[I] := 1;
    for J := I+1 to N do
      if A[I] = A[J] then Inc(B[I]);
    if B[I] > Max then Max := B[I];
  end;
  Write ('MODE(S): ');
  for I := 1 to N do
    if B[I] = Max then Write (A[I], '  ');
  Writeln;
  Writeln ('NUMBER OF OCCURRENCES: ', Max);
end.
```

```
{3.3}
program Thr3T81;
{ -- This program will compute gross weekly pay. }
  var
    E:      String[12];
    R, Pay: Real;
    I:      Byte;
    H:      Array[1..5] of Real;

begin
  Write ('Employee Number: ');  Readln (E);
  Write ('Regular rate of pay/hour: $');  Readln (R);
  Write ('Enter hours for M,T,W,R,F: ');
  Readln (H[1], H[2], H[3], H[4], H[5]);
  Pay := 0;
  for I := 1 to 5 do
    if H[I] <= 8 then
      Pay := Pay + H[I] * R
    else
      Pay := Pay + 8 * R + (H[I]-8) * R * 2;
  Writeln ('EMPLOYEE NUMBER: ', E);
  Writeln ('GROSS WEEKLY PAY: $', Pay :3:2);
end.
```

```
{3.4}
program Thr4T81;
{ -- This program will play tic-tac-toe with a user. }
uses Crt;
  const
    Winsq: Array [1..8,1..3] of Integer =
     { -- Board numbering system }
     ((1,2,3), (8,9,4), (7,6,5),
     { -- Sets of 3 winning squares (in addition to above) }
      (1,8,7), (2,9,6), (3,4,5), (1,9,5), (3,9,7));
    { -- Vertical and horizontal coordinates for squares }
    Row: Array [1..9] of Integer = (1,1,1,3,5,5,5,3,3);
    Col: Array [1..9] of Integer = (1,5,9,9,9,5,1,1,5);
    Pl:  Array [0..1] of String[8] = ('YOU', 'COMPUTER');
    Ast = '  |   |';
    Bst = '---------';
  var
    I, Mov, N, P, X:   Integer;
    A: Array [1..9] of Integer;
```

```
function SomeOneWon: Boolean;
{ -- This procedure checks 8 columns, rows, and diagonals. }
begin
  I := 1;  SomeOneWon := False;
  repeat
    if (A[ Winsq[I,1] ] = P) and (A [Winsq[I,2] ] = P) and
       (A[ Winsq[I,3] ] = P) then
    begin
      GotoXY (3, 10);  Writeln (Pl[P], ' WON!');
      SomeOneWon := True;  I := 8;
    end;
    Inc(I);
  until (I = 9);
end;

begin
  ClrScr;  Writeln (Ast);  Writeln (Bst);  Writeln (Ast);
  Writeln (Bst);  Writeln (Ast);
  for I := 1 to 9 do begin
    A[I] := 9;  GotoXY (Col[I], Row[I]);  Write (I);
  end;
  A[9] := 1;  GotoXY (Col[9], Row[9]);  Write ('X');

  for Mov := 2 to 9 do begin
    if Mov in [2, 4, 6, 8] then P := 0 else P := 1;
    if P = 0 then
      begin          { -- Your move }
        repeat
          GotoXY (3, 8);  Write ('Enter # ');  Readln (N);
          GotoXY (10, 8); Write ('  ');
        until (A[N] <> 0) and (A[N] <> 1);
        A[N] := 0;  GotoXY (Col[N], Row[N]);  Write('O');
        if SomeOneWon then Exit;
      end
    else
      begin          { -- Computer's move }
        repeat
          X := Random (9);
        until (A[X] <> 0) and (A[X] <> 1);
        A[X] := 1;  GotoXY (Col[X], Row[X]);  Write('X');
        if SomeOneWon then Exit;
      end
  end;  { -- for Mov }
  GotoXY (3, 10);  Writeln ('TIE GAME');
end.
```

```
{3.5}
program Thr5T81;
{ -- This program will print a list of people who will retire. }
  const
    TM = 4;  TY = 1981;  { -- Todays month/year }
  var
    I, J, N, Y, Yr, X: Integer;
    Xst:               String[40];
    S, BM, Nam: Array[1..9] of String[18];
    BY:         Array[1..9] of Integer;
    A:          Array[1..9] of Integer;
    B:          Array[1..5,1..9] of Integer;
    SandNam:    Array[1..5,1..9] of String[40];

begin
  Write ('Enter number of employees: ');  Readln (N);
  for I := 1 to N do begin
    Writeln;
    Write ('Social Security No.: ');  Readln (S[I]);
    Write ('Name: ');  Readln (Nam[I]);
    Write ('Birthdate: (Month and day): ');  Readln (BM[I]);
    Write ('Birthdate: (Year): ');  Readln (BY[I]);
  end;
         { -- Determine who retires when }
  for Y := TY - 69 to Ty - 65 do begin
    Yr := Y - (TY - 70);  A[Yr] := 0;
    for I := 1 to N do
      if BY[I] <= Y then begin
        Inc(A[Yr]);
        SandNam[Yr, A[Yr]] := '  #' + S[I] + ' ' + Nam[I];
        B[Yr, A[Yr]] := BY[I];
      end;
  end;
         { -- Dispay retirers }
  for Y := 1 to 5 do
    if A[Y] > 0 then begin { -- sort people by birthdates }
      for I := 1 to A[Y]-1 do
        for J := I+1 to A[Y] do
          if B[Y,I] > B[Y,J] then begin
            X := B[Y,I];  B[Y,I] := B[Y,J];  B[Y,J] := X;
            Xst := SandNam[Y,I];
            SandNam[Y,I] := SandNam[Y, J];
            SandNam[Y,J] := Xst;
          end;
      { -- Display retirers in order of dates }
      Writeln;
      Writeln ('RETIRE WITHIN ', Y, ' YEARS');
      for I := 1 to A[Y] do
        Writeln (SandNam[Y, I]);
    end;
end.
```

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '82 }
{ -- PASCAL PROGRAM SOLUTIONS }


{1.1}
program One1T82;
{ -- This program will allow a user to guess a generated #. }
  var
    X, I, G: Byte;

begin
  Randomize;
  X := Random(100) + 1;  I := 1;
  while (I <= 7) and (G <> X) do begin
    Write ('I AM THINKING OF A NUMBER. WHAT IS IT? ');
    Readln (G);
    if G < X then
      Writeln ('TOO LOW')
    else if G > X then
      Writeln ('TOO HIGH')
    else
      Writeln ('RIGHT');
    Inc(I);
  end;
end.


{1.2}
program One2T82;
{ -- This program will find #s that are the sum of 2 squares. }
  var
    I, J: Byte;
    A:    Array[1..50] of Boolean;

begin
  for I := 1 to 50 do A[I] := False;
  for I := 1 to 5 do
    for J := I to 7 do
      if I*I + J*J < 50 then A[I*I + J*J] := True;

  for I := 1 to 50 do
    if A[I] then Write (I, ',');
  Writeln;
end.
```

```
{1.3}
program One3T82;
{ -- This program will sum numbers divisible by 14. }
  var
    I: Integer;
    S: LongInt;

begin
  for I := 100 to 1000 do
    if I mod 14 = 0 then S := S + I;
  Writeln (S);
end.
```

```
{1.4}
program One4T82;
{ -- This program will add 2 random times. }
  var
    I, M, H:   Byte;
    Min, Hour: Array [1..2] of Byte;

begin
  Randomize;
  for I := 1 to 2 do begin
    Hour[I] := Random(12) + 1;
    Min[I]  := Random(60);
    Write (Hour[I], ':');
    if Min[I] < 10 then Write ('0');
    Writeln (Min[I]);
  end;
  Writeln ('-----');
  M := Min[1] + Min[2];  H := 0;
  if M > 59 then begin
    M := M - 60;  H := 1;
  end;
  H := H + Hour[1] + Hour[2];
  if H > 12 then H := H - 12;
  Write (H, ':');
  if M < 10 then Write ('0');
  Writeln (M);  Writeln;
end.
```

```
{1.5}
program One5T82;
{ -- This program will compute roots of equation. }
  var
    A, B, C, S: Integer;

begin
  Write ('Enter a, b, c: ');  Readln (A, B, C);
  S := B*B - 4*A*C;
  if S < 0 then
    Writeln ('COMPLEX')
  else begin
    Write ( (-B - Sqrt(S)) / (2 * A) : 4:2, '  ');
    Writeln ( (-B + Sqrt(S)) / (2 * A) : 4:2);
  end;
end.
```

```
{1.6}
program One6T82;
{ -- This program will print prime factors. }
  var
    N, I, J: Byte;
    Prime:   Boolean;

begin
  Write ('Enter number: ');  Readln (N);
  for I := 2 to N do
    if N mod I = 0 then begin
      J := 2;  Prime := True;
      while (J <= Trunc(Sqrt(I))) and Prime do begin
        if I mod J = 0 then Prime := False;
        Inc(J);
      end;
      if Prime then Write(I, '  ');
    end;
  Writeln;
end.
```

```
{1.7}
program One7T82;
{ -- This program will calculate future value of investment. }
  var
    P, i:    Real;
    J, N, Y: Integer;

begin
  Write ('Enter P, i, N, Y: ');  Readln (P, i, N, Y);
  for J := 1 to N * Y do
    P := P + P * i / N;
  Writeln ('$', Round(P * 100) / 100 :5:2);
end.
```

```
{1.8}
program One8T82;
{ -- Ths program will find 3 #s whose sum is 43. }
  var
    I, J, K: LongInt;

begin
  for I := 1 to 41 do
    for J := 1 to 42 - I do begin
      K := 43 - I - J;
      if I*I*I + J*J*J + K*K*K = 17299 then begin
        Writeln (I, '  ', J, '  ', K);  Exit;
      end;
    end;
end.


{1.9}
program One9T82;
{ -- This program will print a symbol for 45 seconds. }
uses Crt;
  var
    Ch: Char;

begin
  Write ('Enter a symbol: ');  Readln (Ch);
  ClrScr;  Write(Ch);
  Delay (45000);
  ClrScr;
end.



{1.10}
program One10T82;
{ -- This program will convert decimal to fraction. }
  var
    Dec:              String[12];
    L, N, D, I, Code: Integer;

begin
  Write ('Enter decimal: ');  Readln (Dec);
  L   := Length(Dec) - 1;
  Dec := Copy (Dec, 2, L);
  Val (Dec, N, Code);  D := 1;
  for I := 1 to L do   D := D * 10;
  for I := N downto 1 do
    if (N mod I = 0) and (D mod I = 0) then begin
      Writeln (N div I, '/', D div I);  Exit;
    end;
end.
```

```
{1.11}
program One11T82;
{ -- This program will move an asterisk by pressing keys. }
uses Crt;
  var
    R, C: Integer;
    Ch:   Char;

begin
  ClrScr;  R := 10;  C := 40;
  while Ch <> ' ' do begin
    GotoXY (C, R);  Write ('*');
    Ch := ReadKey;
    if Ch in ['U', 'D', 'L', 'R'] then begin
      GotoXY (C, R);  Write (' ');
      if Ch = 'U' then Dec(R);
      if Ch = 'D' then Inc(R);
      if Ch = 'L' then Dec(C);
      if Ch = 'R' then Inc(C);
    end;
  end;
end.
```

```
{2.1}
program Two1T82;
{ -- This program will print day of week of a date. }
  const
    M: Array [1..12] of Integer =
       (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
  var
    I, Mo, Da, S, X: Integer;

begin
  Write ('Enter month, day: ');  Readln (Mo, Da);
  S := 0;
  for I := 1 to Mo - 1 do S := S + M[I];
  S := S + Da;
  X := S mod 7;
  Writeln ( Copy('THUFRISATSUNMONTUEWED', X*3 + 1, 3) );
end.
```

```
{2.2}
program Two2T82;
{ -- This program will calculate the area of a polygon. }
  var
    N, I, Sum: Integer;
    X, Y:      Array[1..9] of Integer;

begin
  Write ('Enter n: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter vertex (X, Y): ');  Readln (X[I], Y[I]);
  end;
  X[N+1] := X[1];  Y[N+1] := Y[1];  Sum := 0;
  for I := 1 to N do
    Sum := Sum + X[I] * Y[I+1] - Y[I] * X[I+1];
  Writeln ('AREA = ', Abs(Sum) / 2  :4:1);
end.
```

```
{2.3}
program Two3T82;
{ -- This program will find 5 digit number. }
{ -- Strategy:  # is less than 25000 because 4 * # would be
                  a 6 digit # otherwise.
                # can't be 1XXXY since 4 * Y can't give us
                  a 1 in the units place.
                # must therefore begin with 2 and end with
                  8 since 4*8 = 32.  So we can step 10.   }
  var
    I:     LongInt;
    J:     Integer;
    N, S:  String[5];
    Found: Boolean;

begin
  I := 20008;
  repeat
    Str (I, N);  Str (I*4, S);  Found := True;
    for J := 1 to 5 do
      if Copy(N, J, 1) <> Copy(S, 6-J, 1) then
        Found := False;
    if Found then
      Writeln (I)
    else
      I := I + 10;
  until (I >= 24998) or Found;
end.
```

```
{2.4}
program Two4T82;
{ -- This program will find interesting numbers. }
  var
    I, J, K, Num, Pow: Integer;

begin
  for I := 1 to 9 do
    for J := 0 to 9 do
      for K := 0 to 9 do begin
        Num := I * 100 + J * 10 + K;
        Pow := I*I*I + J*J*J + K*K*K;
        if (Num = Pow) and (Num <> 153) then
          Write (Num :5);
      end;
      Writeln;
end.
```

```
{2.5}
program Two5T82;
{ -- This program will make user's name zig zag. }
uses Crt;
  var
    I, X, L, S: Byte;
    Nam:        String[20];
    Ch:         Char;

begin
  Write ('Enter name: ');  Readln (Nam);  ClrScr;
  L := Length (Nam);
  X := Trunc(159 / (L-1));
  for I := 1 to L do begin
    Ch := Nam[I];
    S := (I - 1) * X;
    if S > 79 then S := 159 - S;
    Writeln (' ': S, Ch);
  end;
end.

{2.6}
program Two6T82;
{ -- This program will print a stick figure. }
uses Crt;
  var
    R, C, I, K: Byte;
    Inc:        Real;
    A:          Char;

begin
  R := 5;  C := 12;
  repeat
    for I := 0 to 5 do begin
      ClrScr;
      Writeln (' *     ***** ');
      Writeln ('  *    *   * ');
      Writeln ('   *   ***** ');
      Writeln ('    **   *   ');
      Writeln ('      *****  ');
      Writeln ('         *   ');
      Writeln ('         *   ');
      Writeln ('        * *  ');
      Writeln ('       *   * ');
      Writeln ('      *     *');
      Inc := (R - I) / 7;
      For K := 0 to 6 do begin
        GotoXY (C+K, R-Trunc(Inc*K));  Write ('*');
      end;
      Delay(100);
    end;
    A := ReadKey;
  until A = Char(27);
end.
```

```
{2.7}
program Two7T82;
{ -- This program will display permutations of letters. }
uses Crt;
  var
    N, I, X: Integer;
    A:       Array [1..8] of Char;
    Temp:    Char;

begin
  Randomize;
  Write ('How many letters: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter letter: ');  Readln (A[I]);
  end;
  repeat
    for I := 1 to N do begin
      X := Random(N) + 1;
      Temp := A[X];  A[X] := A[I];  A[I] := Temp;
    end;
    for I := 1 to N do Write (A[I]);
    Writeln; Delay(100);
  until Keypressed;
end.
```

```
{2.8}
program Two8T82;
{ -- This program will drill typying skills. }
uses Crt;
  var
    I, X, J: Integer;
    S:       LongInt;
    A, B:    Array[1..4] of Char;
    Ch:      Char;
    Wrong:   Boolean;

begin
  Randomize;
  for I := 1 to 4 do begin
    X := Random(58) + 33;
    A[I] := Chr(X);
    Write (A[I], ' ');
  end;
  Writeln;  J := 1;  S := 0;
  while J < 5 do begin
    repeat
      Inc(S);
    until Keypressed;
    Ch := ReadKey;  B[J] := Ch;
    Write (Ch, ' ');
    Inc(J);
  end;
  Writeln;  Writeln;  Wrong := False;
  for I := 1 to 4 do
    if A[I] <> B[I] then begin
      Writeln (A[I], ' --- ', B[I], '  NO');
      Wrong := True;
    end;
  if Not Wrong then Writeln (S div 30000, ' SECONDS');
end.
```

```
{2.9}
program Two9T82;
{ -- This program will return change in fewest coins. }
  const
    Nam: Array [1..8] of String[9] =
      ('$20', '$10', '$5', 'DOLLARS', 'QUARTERS', 'DIMES',
       'NICKELS', 'PENNIES');
    Amount: Array [1..8] of Integer =
      (2000, 1000, 500, 100, 25, 10, 5, 1);
  var
    P: Real;
    N, D, I, X: Integer;

begin
  Write ('Enter price $: ');  Readln (P);
  Write ('Enter denomination $: ');  Readln (D);
  N := D * 100 - Trunc(P * 100 + 0.1);
  for I := 1 to 8 do begin
    X := N div Amount[I];
    if X > 0 then Writeln (X, ' ', Nam[I]);
    N := N - X * Amount[I];
  end;
end.
```

```
{2.10}
program Two10T82;
{ -- This program will make unit conversions. }
  const
    A: Array[1..5] of String[2] =
      ('IN', 'FT', 'FT', 'YD', 'MI');
    B: Array[1..5] of String[2] =
      ('CM', 'CM', 'M ', 'M ', 'KM');
  var
    I, X: Byte;
    N, S: Real;

begin
  for I := 1 to 5 do
    Writeln (I, '  ', A[I], ' -> ', B[I]);
  Write ('Enter Choice #: ');  Readln (X);
  Write ('Enter ', A[X], ': ');  Readln (N);
  S := N * 2.54;
  if X = 1 then Write (S :6:2);
  if X = 2 then Write (S * 12 :6:2);
  if X = 3 then Write (S * 12 / 100 :6:2);
  if X = 4 then Write (S * 36 / 100 :6:2);
  if X = 5 then Write (S * 5280 * 12 / 100000.0 :6:2);
  Writeln (' ', B[X]);
end.
```

```
{2.11}
program Two11T82;
{ -- This program will find A^B x C^D = ABCD }
  var
    A, B, C, D, J, APow, CPow, Num: LongInt;

begin
  for A := 1 to 9 do
    for B := 0 to 9 do
      for C := 0 to 9 do
        for D := 0 to 9 do begin
          APow := 1;  CPow := 1;
          for J := 1 to B do APow := APow * A;
          for J := 1 to D do CPow := CPow * C;
          Num := A*1000 + B*100 + C*10 + D;
          if APow * CPow = Num then begin
            Writeln ('A=', A, '  B=', B, '  C=', C, '  D=',D);
            Exit;
          end;
        end;
end.
```

```
{2.12}
program Two12T82;
{ -- This program calculates days between 2 dates. }
  const
    Days: Array[1..12] of Integer =
      (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
  var
    M1, D1, M2, D2, I, S: Integer;

begin
  Write ('Enter Month1, Day1: ');  Readln (M1, D1);
  Write ('Enter Month2, Day2: ');  Readln (M2, D2);
  S := 0;
  for I := M1 to M2-1 do S := S + Days[I];
  Writeln (S + D2 - D1, ' DAYS');
end.
```

```
{2.13}
program Two13T82;
{ -- This program will print a check. }
uses Crt;
  const
    Mo: Array [1..12] of String[5] =
      ('JAN.', 'FEB.', 'MAR.', 'APRIL', 'MAY', 'JUNE',
       'JULY', 'AUG.', 'SEPT.', 'OCT.', 'NOV.', 'DEC.');
    Words: Array[1..27] of String[10] =
      ('ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN',
       'EIGHT', 'NINE', 'TEN', 'ELEVEN', 'TWELVE', 'THIRTEEN',
       'FOURTEEN', 'FIFTEEN', 'SIXTEEN', 'SEVENTEEN',
       'EIGHTEEN', 'NINETEEN', 'TWENTY-', 'THIRTY-', 'FOURTY-',
       'FIFTY-', 'SIXTY-', 'SEVENTY-', 'EIGHTY-', 'NINETY-');
  var
    I, M, D, Y, S, T, X, Cent: Integer;
    Nam: String[20];
    N:   Real;

begin
  Write ('Enter month, day, year: ');  Readln (M, D, Y);
  Write ('Enter amount $:');  Readln (N);
  Write ('Enter payee: ');    Readln (Nam);
       { -- Display check border }
  ClrScr;
  for I := 1 to 60 do Write ('*');
  for I := 1 to 7 do begin
    GotoXY (1, I+1);  Write ('*');
    GotoXY (60,I+1);  Write ('*');
  end;
  Writeln;
  for I := 1 to 60 do Write ('*');
       { -- Display date, Name, and amount }
  GotoXY (45, 2);  Write (Mo[M], ' ', D, ', 19', Y);
  GotoXY (5, 4);   Write ('PAY TO THE');
  GotoXY (5, 5);   Write ('ORDER OF  ', Nam);
  GotoXY (50, 5);  Write ('$', N:5:2);
  GotoXY (3, 7);
       { -- Display amount in words }
  Cent := Trunc( (N - Int(N)) * 100 + 0.1);  S := 1000;  T := 0;
  for I := 2 downto 0 do begin
    S := S div 10;  X := Trunc(N/S + 0.001);
    if (I = 2) and (X > 0) then
      Write (Words[X], ' HUNDRED ');
    if (I = 1) and (X > 1) then
      Write (Words[18+X]);
    if (I = 1) and (X = 1) then T := 1 else T := 0;
    if I = 0 then
      Write (Words[T*10+X]);
    N := Int(N - X * S + 0.001);
  end;
  Write (' AND ', Cent, '/100 DOLLARS');
end.
```

```
{3.1}
program Thr1T82;
{ -- This program will play mastermind. }
{ -- The computer will randomly select four of the six colors.  }
{ -- The user must guess this combination of four colors.       }
{ -- BLACK indicates that a color is in the right position.     }
{ -- WHITE indicates a color is right but in the wrong position.}
uses Crt;
  const
    Co: Array [1..6] of String[2] =
      ('W', 'Y', 'R', 'G', 'BL', 'BK');
  var
    I, J, K, W, Bk, X: Integer;
    A, B, C:           Array[1..6] of String[2];

begin
  Randomize;
  for I := 1 to 4 do begin
    X := Random(6) + 1;  A[I] := Co[X];
  end;

  ClrScr;  Writeln ('GUESS: W, Y, R, G, BL, BK');
  for K := 1 to 10 do begin
    W := 0;  Bk := 0;
    for I := 1 to 4 do begin
      GotoXY (I*6, K*2);  Readln (B[I]);
    end;
    for I := 1 to 4 do C[I] := A[I];
    for I := 1 to 4 do
      if C[I] = B[I] then begin
        Inc(Bk);  B[I] := '';  C[I] := ' ';
      end;
    for I := 1 to 4 do
      for J := 1 to 4 do
        if C[I] = B[J] then begin
          Inc(W);  B[J] := '';  C[I] := ' ';
        end;
    { -- Black pegs = Correct color and correct position }
    { -- White pegs = Correct color but wrong position }
    GotoXY (40, K*2);
    Write ('BLACKS = ', Bk, '   WHITES = ', W);
    if Bk = 4 then begin
       Writeln; Writeln ('YOU WIN IN ', K, ' TURNS');  Exit;
    end;
  end;  { -- for K }
  Writeln;  Writeln ('YOU LOSE');
  for I := 1 to 4 do Write (A[I], ' ');
end.
```

```
{3.2}
program Thr2T82;
{ -- This program will plot points on a new axis. }
uses Crt;
  var
    X1, Y1, X2, Y2, IT, N, I, R, C: Integer;
    X, Y:                           Array[1..9] of Integer;

begin
  Write ('Enter end point of x-axis: ');  Readln (X1, Y1);
  Write ('Enter end point of y-axis: ');  Readln (X2, Y2);
  Write ('Enter increment: ');  Readln (IT);
  Write ('How many points: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter point: ');  Readln (X[I], Y[I]);
  end;
  ClrScr;  R := 3;  C := 1;
  Writeln ('INTERSECTION AT (', X2, ',', Y1, ')');
  Writeln;
  I := Y1;
  repeat
    Write ('*');  I := I + IT;
  until I > Y2;
  I := X2 + IT; Writeln;
  repeat
    Writeln ('*');  I := I + IT;
  until I > X1;
  for I := 1 to N do begin
    GotoXY (C + (Y[I]-Y1) div IT, R + (X[I]-X2) div IT);
    Write ('+');
  end;
end.
```

```
{3.3}
program Thr3T82;
{ -- This program will generate magic squares. }
{ -- Correctly for odd matrices and for a 4 x 4. }
uses Crt;
  var
    N, X, Y, I, J, S: Integer;
    A: Array [1..12, 1..12] of Integer;

begin
  ClrScr;
  Write ('Enter size: ');  Readln (N);
  Writeln;  S := 0;
  if N mod 2 = 1 then begin  { -- routine for odd Matrix }
    for X := 1 to N do
      for Y := 1 to N do
        A[X,Y] := 0;
    X := 1;  Y := (N+1) div 2;  A[X,Y] := 1;
    for I := 2 to N*N do begin
      Dec(X);  Dec(Y);
      if X = 0 then X := N;
      if Y = 0 then Y := N;
      if A[X,Y] = 0 then
        A[X,Y] := I
      else begin
        X := X + 2;  Inc(Y);
        if X > N then X := X - N;
        if Y > N then Y := 1;
        A[X,Y] := I;
      end;
    end;  { -- for I }
    end   { -- begin }
  else    { -- Routine for Even Matrix (4x4) }
    for I := 1 to N do
      for J := 1 to N do begin
        S := S + 1;
        if (I = J) or (I = N+1-J) then
          A[I,J] := S
        else
          A[I,J] := N*N + 1 - S;
      end;

  for I := 1 to N do
    for J := 1 to N do begin
      GotoXY (J*4, I*2);  Write (A[I,J]);
    end;
  Writeln;  Writeln ('MAGIC NUMBER = ', (N*N*N + N) div 2);
end.
```

```
{3.4}
program Thr4T82;
{ -- This program will add and multiply 2 Roman Numerals. }
  const
    RN: Array[1..7] of Char =
      ('M', 'D', 'C', 'L', 'X', 'V', 'I');
    RV: Array[1..7] of Integer =
      (1000, 500, 100, 50, 10, 5, 1);
  var
    I, E, L, Ar, I1, I2, J, K, XX, Num: Integer;
    Rom, R:  Array [1..2] of String[15];
    A, N:    Array [1..2] of Integer;
    Ch, NCh: String[1];
    X:       Real;

begin
  for E := 1 to 2 do begin
    Write ('Enter Roman Numeral: ');  Readln (Rom[E]);
    L := Length(Rom[E]);  I := 1;  Ar := 0;
    while I < L do begin
      Ch := Copy (Rom[E], I, 1);
      I1 := 1;  while Ch <> RN[I1] do Inc(I1);
      NCh:= Copy (Rom[E], I+1, 1);
      I2 := 1;  while NCh <> RN[I2] do Inc(I2);
      if I1 <= I2 then
        Ar := Ar + RV[I1]
      else begin
        Ar := Ar + RV[I2] - RV[I1];  Inc(I);  end;
      Inc(I);
    end;
    if I <= L then begin  { -- Last numeral not done }
      Ch := Copy (Rom[E], I, 1);
      I1 := 1;  while Ch <> RN[I1] do Inc(I1);
      Ar := Ar + RV[I1];
    end;
    A[E] := Ar;
  end;  { -- for E }
  { -- Convert Arabic numbers to Roman Numerals }
  N[1] := A[1] + A[2];  N[2] := A[1] * A[2];
  R[1] := '';  R[2] := '';
  for K := 1 to 2 do begin
    Num := N[K];
    for I := 1 to 7 do begin
      X := Num / RV[I];
      if (X<2) and (X>=9/5) and (I in [2,4,6]) then { -- next }
      else begin
        XX := Trunc(X);
        if XX = 9 then R[K] := R[K] + RN[I] + RN[I-2]
        else
        if XX = 4 then R[K] := R[K] + RN[I] + RN[I-1]
        else
        if XX > 0 then
          for J := 1 to XX do
            R[K] := R[K] + RN[I];
        Num := Num - RV[I] * XX;
      end;
    end;  { -- for I }
  end;  { -- for K }
  { -- Display sum and product }
  Writeln (Rom[1], ' + ', Rom[2], ' = ', R[1]);
  Writeln (A[1], ' + ', A[2], ' = ', N[1]);
  Writeln (Rom[1], ' * ', Rom[2], ' = ', R[2]);
  Writeln (A[1], ' * ', A[2], ' = ', N[2]);
end.


{3.5}
program Thr5T82;
{ -- This program will find 4 digit squumbers. }
  var
    I, L, R, X, Code: Integer;
    Ist:              String[4];

begin
  for I := 1000 to 9999 do begin
    Str (I, Ist);
    Val (Copy(Ist, 1, 2), L, Code);
    Val (Copy(Ist, 3, 2), R, Code);
    X := L + R;
    if X * X = I then Writeln (I);
  end;
end.

{3.6}
program Thr6T82;
{ -- This program should play Nim with a user.          }
{ -- HOWEVER, since the rules are not given with this    }
{ -- problem, it is very difficult to write the program. }
begin
end.


{3.7}
program Thr7T82;
{ -- This program will determine where a # falls in a list. }
  var
    A:      Array [1..16] of Integer;
    I, Num: Integer;

begin
  for I := 1 to 16 do begin
    Write ('Enter #: ');  Readln (A[I]);
  end;
  Write ('Enter another number: ');  Readln (Num);
  I := 1;
  while A[I] <> Num do Inc(I);
  Writeln ('BETWEEN ', A[I-1], ' AND ', A[I+1]);
end.
```

```
{3.8}
program Thr8T82;
{ -- This BONUS program will guess the user's state. }
  const
    State: Array[1..50] of String[14] =
    ('ALABAMA','ALASKA','ARIZONA','ARKANSAS','CALIFORNIA',
    'COLORADO','CONNECTICUT','DELEWARE','FLORIDA','GEORGIA',
    'HAWAII','IDAHO','ILLINIOS','INDIANA','IOWA','KANSAS',
    'KENTUCKY','LOUISIANA','MAINE','MARYLAND','MASSACHUSETTS',
    'MICHIGAN','MINNESOTA','MISSISSIPPI','MISSOURI','MONTANA',
    'NEBRASKA','NEVADA','NEW HAMPSHIRE','NEW JERSEY','NEW YORK',
    'NEW MEXICO','NORTH CAROLINA','NORTH DAKOTA','OHIO',
    'OKLAHOMA','OREGON','SOUTH CAROLINA','SOUTH DAKOTA',
    'PENNSYLVANIA','RHODE ISLAND','TENNESSEE','TEXAS','UTAH',
    'VERMONT','VIRGINIA','WASHINGTON','WEST VIRGINIA',
    'WISCONSIN','WYOMING');
  var
    I, G, B, M, E: Integer;
    A:             String[3];

begin
  G := 1;  B := 1;  M := 25;  E := 50;
  repeat
    Write (G, '- IS YOUR STATE ALPHABETICALLY BEFORE ', State[M]);
    Writeln;
    Write ('Enter YES or NO: ');  Readln (A);
    if (A = 'YES') and (B+1 = M) then begin
      Writeln (State[B], ' IS IT');  Exit;  end;
    if (A = 'NO') and (M = E) then begin
      Writeln (State[M], ' IS IT');  Exit;  end;
    if A = 'YES' then begin
      E := M - 1;  M := M - Round((M - B) / 2);  end
    else begin
      B := M;      M := M + Round((E - M) / 2);  end;
    Inc(G);
  until G > 12;
end.
```

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '83 }
{ -- PASCAL PROGRAM SOLUTIONS }


{1.1}
program One1T83;
{ -- This program will round a number to nearest whole number. }
  var
    Num: Real;

begin
  Write ('Enter number: ');  Readln (Num);
  Writeln (Round(Num));
end.


{1.2}
program One2T83;
{ -- This program will display 5 numbers in descending order. }
  var
    I, J, X: Integer;
    A:       Array [1..5] of Integer;

begin
  for I := 1 to 5 do begin
    Write ('Enter number: ');  Readln (A[I]);
  end;
  for I := 1 to 4 do
    for J := I+1 to 5 do
      if A[I] < A[J] then begin
        X := A[I];  A[I] := A[J];  A[J] := X;
      end;

  for I := 1 to 5 do
    Writeln (A[I]);
end.


{1.3}
program One3T83;
{ -- This program will print the factors of a given number. }
  var
    Num, I: Integer;

begin
  Write ('Enter number: ');  Readln (Num);
  for I := 1 to Num do
    if Num mod I = 0 then
      Writeln (I);
end.
```

```
{1.4}
program One4T83;
{ -- This program will produce a birthday card w/name centered. }
  var
    I, J, L, Sp: Integer;
    Name:        String[10];

begin
  Write ('Enter name: ');  Readln (Name);
  for I := 1 to 5 do begin
    Writeln;
    if I in [1, 5] then
      for J := 1 to 12 do
        Write ('*')
    else if (I = 2) then
      Write ('*   HAPPY  *')
    else if (I = 3) then
      Write ('* BIRTHDAY *')
    else begin
      Write ('*');
      L := Length(Name);
      Sp := (11-L) div 2;
      Write (' ': Sp, Name, ' ': 10-L-Sp, '*');
    end;
  end;
end.
```

```
{1.6}
program One6T83;
{ -- This program will print a B for A, C for B, ... Z for A. }
  var
    Ch: Char;

begin
  Write ('Enter Character: ');  Readln (Ch);
  if Ch < 'Z' then
    Writeln (Char(Ord(Ch) + 1))
  else  { -- Z was entered }
    Writeln ('A');
end.
```

```
{1.5}
program One5T83;
{ -- This program will print a ? in random locations. }
uses Crt;
  var
    I, X, Y: Byte;

begin
  ClrScr;  Randomize;
  for I := 1 to 6 do begin
    X := Random(80) + 1;  Y := Random(24) + 1;
    GotoXY (X, Y);  Write ('?');
    Delay (4000);
  end;
end.
```

```
{1.7}
program One7T83;
{ -- This program will print 4 distinct rectangles in corners. }
uses Crt;

procedure Rectangle ({At} Row, Col: Integer);
{ -- This procedure will produce a 10 by 4 rectangle at X, Y }
  var
    I, J: Byte;

begin
  for I := Row to Row+3 do
    if (I = Row) or (I = Row+3) then begin
      GotoXY (Col, I);
      for J := 1 to 10 do
        Write ('*');
      end
    else begin
      GotoXY (Col, I);   Write ('*');
      GotoXY (Col+9, I); Write ('*');
    end;
end;

begin
  ClrScr;
  Rectangle (1, 1);
  Rectangle (1, 65);
  Rectangle (19, 1);
  Rectangle (19, 65);
end.
```

```
{1.8}
program One8T83;
{ -- This program will count the number of e's in a sentence. }
  var
    Sent: String[80];
    I, E: Byte;
    Ch:   Char;

begin
  Write ('Enter sentence: ');  Readln (Sent);
  E := 0;
  for I := 1 to Length(Sent) do begin
    Ch := UpCase( Sent[I] );
    if Ch = 'E' then Inc(E);
  end;
  Writeln (E, ' E''s');
end.
```

```
{1.9}
program One9T83;
{ -- This program will calculate the average score for a person.}
  const
    Name:   Array [1..3] of String[4] = ('JOHN', 'BILL', 'MARY');
    Scores: Array [1..3,1..3] of Byte =
      ((20, 70, 32),  (71, 40, 30),  (80, 42, 73));
  var
    I, J, Total, Ind: Byte;
    St:               String[4];

begin
  Write ('Enter name: ');  Readln (St);
  for I := 1 to 3 do
    if St = Name[I] then Ind := I;
  Total := 0;
  for J := 1 to 3 do
    Total := Total + Scores[Ind, J];
  Writeln ('Average = ', Total / 3 :3:2);
end.
```

```
{1.10}
program One10T83;
{ -- This program will reverse the digits of a 4 digit number. }
  var
    I:  Byte;
    St: String[4];

begin
  Write ('Enter number: ');  Readln (St);
  for I := 4 downto 1 do
    Write (Copy(St, I, 1));
  Writeln;
end.
```

```
{2.1}
program Two1T83;
{ -- This program will calculate the area of a regular hexagon. }
  var
    Perim, S: Real;

begin
  Write ('Enter perimeter: ');  Readln (Perim);
  S := Perim / 6;
  Writeln ( (Sqrt(3)*S/2 * S/2) * 6 :7:4);
end.
```

```
{2.2}
program Two2T83;
{ -- This program will convert a base 8 num to a base 2 num. }
  var
    I, Digit:  Byte;
    FirstDig:  Byte;
    Ch:        Char;
    Num:       String[4];
    St:        String[12];

begin
  Write ('Enter number: ');  Readln (Num);
  St := '';
  for I := 1 to Length(Num) do begin
    Ch := Num[I];
    Digit := Ord(Ch) - Ord('0');
    case Digit of
      0: St := St + '000';
      1: St := St + '001';
      2: St := St + '010';
      3: St := St + '011';
      4: St := St + '100';
      5: St := St + '101';
      6: St := St + '110';
      7: St := St + '111';
    end;
  end;
  FirstDig := 1;
  while Copy(St, FirstDig, 1) = '0' do
    Inc(FirstDig);
  Writeln (Copy(St, FirstDig, Length(St)-FirstDig+1));
end.
```

```
{2.3}
program Two3T83;
{ -- This program will add several items with tax (5%). }
  var
    Item, Tax, Total: Real;

begin
  Total := 0;
  Write ('Enter item: ');  Readln (Item);
  while Item <> -999 do begin
    Total := Total + Item;
    Write ('Enter item: ');  Readln (Item);
  end;
  Writeln ('SUBTOTAL = $', Total: 5:2);
  Tax := Total * 0.05;
  Writeln ('TAX      = $', Tax:   5:2);
  Total := Total + Tax;
  Writeln ('TOTAL    = $', Total: 5:2);
end.
```

```
{2.4}
program Two4T83;
{ -- This program will divide the screen into 4 rectangles. }
uses Crt;
  var
    Ch:   Char;
    I, J: Integer;

begin
  Write ('Enter character: ');  Readln (Ch);
  ClrScr;
  for I := 1 to 24 do
    if I <> 12 then
      Writeln (' ': 39, Ch)
    else
      for J := 1 to 80 do
        Write (Ch);
end.
```

```
{2.5}
program Two5T83;
{ -- This program will print the greatest and least in a set. }
  var
    Max, Min, Num: Real;

begin
  Max := -900;  Min := 900;
  Write ('Enter number: ');  Readln (Num);
  while Num <> -999 do begin
    if Num < Min then Min := Num
    else if Num > Max then Max := Num;
    Write ('Enter number: ');  Readln (Num);
  end;
  Writeln ('GREATEST = ', Max :5:2);
  Writeln ('LEAST = ', Min :5:2);
end.
```

```
{2.6}
program Two6T83;
{ -- This program will print the sum, mean, median. }
  var
    I, J:   Byte;
    Sum, X: Real;
    A:      Array [1..10] of Real;

begin
  Sum := 0;
  for I := 1 to 10 do begin
    Write ('Enter number: ');  Readln (A[I]);
    Sum := Sum + A[I];
  end;
  { -- Sort 10 numbers }
  for I := 1 to 9 do
    for J := I+1 to 10 do
      if A[I] > A[J] then begin
        X := A[I];  A[I] := A[J];  A[J] := X;
      end;

  Writeln ('SUM = ', Sum: 5:2);
  Writeln ('MEAN = ', Sum / 10 :5:2);
  Writeln ('MEDIAN = ', (A[5] + A[6])/2 :5:2);
end.
```

```
{2.7}
program Two7T83;
{ -- This program will reverse the words in a sentence. }
{ -- Assume 1 space between each word. }
  var
    Sent:   String[80];
    Word:   Array [1..10] of String[10];
    I, Num: Byte;
    Ch:     Char;

begin
  Write ('Enter sentence: ');  Readln (Sent);
  Num := 1;  Word[Num] := '';
  for I := 1 to Length(Sent) do begin
    Ch := Sent[I];
    if Ch <> ' ' then
      Word[Num] := Word[Num] + Ch
    else begin
      Inc(Num);
      Word[Num] := '';
    end;
  end;
  for I := Num downto 1 do
    Write (Word[I], ' ');
  Writeln;
end.
```

```
{2.8}
program Two8T83;
{ -- This program will convert cubic feet to cubic meters. }
{ -- (1 in. = 2.54 cm) }
  var
    CF, CM, CM3: Real;

begin
  Write ('Enter cubic feet: ');  Readln (CF);
  CM3 := CF * (12 * 2.54) * (12 * 2.54) * (12 * 2.54);
  CM := CM3/ 100 / 100 / 100;
  Writeln (CM :7:4, ' CUBIC METERS');
end.
```

```
{2.9}
program Two9T83;
{ -- This program will find sum of Ys for Xs for Y=2(X+5). }
  var
    A, B, X, Sum: Integer;

begin
  Write ('Enter a and b: ');  Readln (A, B);  Sum := 0;
  for X := A to B do
    Sum := Sum + 2 * (X+5);
  Writeln ('SUM = ', Sum);
end.
```

```
{2.10}
program Two10T83;
{ -- This program will print 1 char. for 10 sec, 2 for 10 sec.. }
uses Crt;
  var
    I, J: Byte;
    Ch:   Char;

begin
  Write ('Enter character: ');  Readln (Ch);  ClrScr;
  for I := 1 to 10 do begin
    for J := 1 to I do
      Write (Ch);
    Delay (5000);
    ClrScr;  Delay (500);
  end;
end.
```

```
{3.1}
program Thr1T83;
{ -- This program converts a number for one base to another. }
  var
    Base1, Base2, Num1V, Num2, Power: Integer;
    I, J, K, X, Digit:                Integer;
    Num1:                             String[7];
    Ch:                               Char;

begin
  Write ('ENTER NUMBER: ');  Readln (Num1);
  Write ('ENTER BASE: ');    Readln (Base1);
  Write ('CONVERT TO BASE: '); Readln (Base2);
  Write ('ANSWER IS ');

  { -- Convert Num1 to base 10 number Num1V }
  Num1V := 0;
  for I := 1 to Length(Num1) do begin
    Ch := Num1[I];
    Digit := Ord(Ch) - Ord('0');
    Power := 1;
    for J := 1 to Length(Num1) - I do
      Power := Power * Base1;
    Num1V := Num1V + Digit * Power;
  end;

  { -- Convert Num1V to Base2 number }
  J := Trunc(Ln(Num1V) / Ln(Base2));
  for I := J downto 0 do begin
    Power := 1;
    for K := 1 to I do  Power := Power * Base2;
    X := Num1V div Power;
    Write (X);
    Num1V := Num1V - X * Power;
  end;
  Writeln;
end.
```

```
{3.2}
program Thr2T83;
{ -- This program determines what triangle is made w/3 points. }
  var
    X1, Y1, X2, Y2, X3, Y3: Integer;
    D1, D2, D3:             Real;

begin
  Write ('Enter X1, Y1: ');  Readln (X1, Y1);
  Write ('Enter X2, Y2: ');  Readln (X2, Y2);
  Write ('Enter X3, Y3: ');  Readln (X3, Y3);

  { -- Calculate distances }
  D1 := Sqrt ((X1-X2)*(X1-X2) + (Y1-Y2)*(Y1-Y2));
  D2 := Sqrt ((X2-X3)*(X2-X3) + (Y2-Y3)*(Y2-Y3));
  D3 := Sqrt ((X3-X1)*(X3-X1) + (Y3-Y1)*(Y3-Y1));

  { -- No triangle can be formed if sum of 2 sides equals third. }
  if (D1+D2 = D3) or (D1+D3 = D2) or (D2+D3 = D1) then
    Writeln ('NOT A TRIANGLE')
  else if (D1 = D2) and (D2 = D3) then
    Writeln ('EQUILATERAL')
  else if (D1 = D2) or (D2 = D3) or (D1 = D3) then
    Writeln ('ISOSCELES')
  else
    Writeln ('SCALENE');
end.
```

```
{3.3}
program Thr3T83;
{ -- This program randomly selects an X, Y in 10 x 10 grid. User
  -- guesses numbers; if guess is wrong, a direction is given. }

  var
    X, Y, A, B: Byte;

begin
  Randomize;
  X := Random(10) + 1;  Y := Random(10) + 1;
  repeat
    Write ('Enter X, Y: ');  Readln (A, B);
         if (A = X) and (B < Y) then Writeln ('UP')
    else if (A = X) and (B > Y) then Writeln ('DOWN')
    else if (A > X) and (B = Y) then Writeln ('LEFT')
    else if (A < X) and (B = Y) then Writeln ('RIGHT')
    else if (A < X) and (B < Y) then Writeln ('UP AND RIGHT')
    else if (A < X) and (B > Y) then Writeln ('DOWN AND RIGHT')
    else if (A > X) and (B < Y) then Writeln ('UP AND LEFT')
    else if (A > X) and (B > Y) then Writeln ('DOWN AND LEFT');
  until (A=X) and (B=Y);
end.
```

```
{3.4}
program Thr4T83;
{ -- This program will divide 1st number by 2nd out to N places. }
  var
    Num1, Num2, Places, I, X: Integer;

begin
  Write ('ENTER FIRST NUMBER: ');  Readln (Num1);
  Write ('ENTER SECOND NUMBER: '); Readln (Num2);
  Write ('ENTER NUMBER OF DECIMAL PLACES: ');  Readln (Places);
  Write ('QUOTIENT IS ');
  X := Num1 div Num2;  Write (X, '.');
  Num1 := Num1 - Num2*X;
  for I := 1 to Places do begin
    Num1 := Num1 * 10;
    X := Num1 div Num2;
    Write (X);
    Num1 := Num1 - Num2*X;
  end;
end.
```

```
 {3.5}
program Thr5T83;
{ -- 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). }
uses Crt;
  var
    I, J, X, R1, R2, IndX, IndY: Byte;
    Digit, BlankX, BlankY:       Byte;
    A:                           Array [1..3, 1..3] of Byte;
    Valid:                       Boolean;
    DigSt:                       String[1];
    Code:                        Integer;
```

```
begin
  { -- Randomly place numbers in Array A. }
  Randomize;
  for I := 1 to 3 do
    for J := 1 to 3 do
      A[I,J] := (I-1)*3 + J-1;
  for I := 1 to 3 do
    for J := 1 to 3 do begin  { -- swap array values }
      R1 := Random(3) + 1;  R2 := Random(3) + 1;
      X := A[I,J];  A[I,J] := A[R1,R2];  A[R1,R2] := X;
    end;
  repeat
    { -- Display array }
    ClrScr;
    for I := 1 to 3 do begin
      for J := 1 to 3 do
        if A[I,J] > 0 then Write (A[I,J], '  ')
        else begin
          Write ('   ');
          BlankX := I;  BlankY := J;
        end;
      Writeln;
    end;

    { -- Accept valid digit or 9 }
    Valid := False;
    repeat
      DigSt := ''; while DigSt = '' do DigSt := ReadKey;
      Val(DigSt,Digit,Code);
      for I := 1 to 3 do
        for J := 1 to 3 do
          if Digit = A[I,J] then begin
            IndX := I;  IndY := J;
          end;
      if Abs(BlankX - IndX) + Abs(BlankY - IndY) = 1 then
        { -- adjacent }
        Valid := True;
    until Valid or (Digit = 9);

    if Valid then begin  { -- move digit in space }
      X := A[IndX,IndY];  A[IndX,IndY] := A[BlankX,BlankY];
      A[BlankX,BlankY] := X;
    end;
  until Digit = 9;  { -- 9 pressed }
end.
```

```
{3.6}
program Thr6T83;
{ -- This program will store a list of words and provide options.}
  var
    Option, I, J, Num: Byte;
    Word:              Array [1..10] of String[10];
    DeleteW:           String[10];

begin
  Num := 0;
  repeat
    Writeln;
    Writeln ('1. ADD A WORD TO THE LIST');
    Writeln ('2. DELETE A WORD FROM THE LIST');
    Writeln ('3. DISPLAY THE ENTIRE LIST');
    Readln (Option);
    case Option of
      1: begin
           Inc(Num);
           Write ('Enter word: ');  Readln (Word[Num]);
         end;
      2: begin
           Write ('Enter word: ');  Readln (DeleteW);
           I := 1;
           while (I <= Num) and (Word[I] <> DeleteW) do
             Inc(I);
           for J := I to Num-1 do Word[J] := Word[J+1];
           Dec(Num);
         end;
      3: for I := 1 to Num do
           Writeln (Word[I]);
    end;
  until Option > 3;
end.
```

```
{3.7}
program Thr7T83;
{  --  This  program  will  solve  cryptorithms  with  two  2-letter
addends
  -- and a 3-letter sum, using only the letters A,B,C,D, and E. }

  var
    St1, St2, St3:       String[3];
    Letters, Numbers:    String[7];
    FirstLet, UniqueLet: Array [1..7] of Integer;
    N1St, N2St, SumSt:   String[3];
    Ch:                  Char;
    Solution, AtLeast1:  Boolean;
    I, J, N1, N2, Sum, NumLet: Integer;
```

```
begin
  Write ('Enter FIRST ADDEND: ');  Readln (St1);
  Write ('Enter SECOND ADDEND: '); Readln (St2);
  Write ('Enter SUM: ');           Readln (St3);
  Letters := St1 + St2 + St3;  NumLet := 0;  AtLeast1 := False;

  { Put in FirstLet[] the index of the first occurence of letter }
  for I := 1 to 7 do begin
    Ch := Letters[I];
    FirstLet[I] := Pos(Ch, Letters);
    if FirstLet[I] = I then begin { -- This is a new letter. }
      Inc(NumLet);
      UniqueLet[NumLet] := I;
    end;
  end;

  for N1 := 10 to 98 do             { -- N1 must be 2 digits, >9 }
    for N2 := 100-N1 to 98 do begin { -- N2 must be 2 digits, >9 }
      Sum := N1 + N2;               { -- Sum must be 3 digits,>99}
      Str (N1, N1St);  Str (N2, N2St);  Str (Sum, SumSt);
      Numbers := N1St + N2St + SumSt;
      I := 1;  Solution := True;
      { -- Check if similar letters correspond to similar numbers}
      repeat
        Ch := Numbers[I];
        if Ch <> Copy (Numbers, FirstLet[I], 1) then
          Solution := False;
        Inc(I);
      until (I > 7) or not Solution;

      { -- Check if unique letters correspond to unique digits }
      for I := 1 to NumLet-1 do
        for J := I+1 to NumLet do
          if Numbers[UniqueLet[I]] = Numbers[UniqueLet[J]] then
            Solution := False;

      if Solution then begin  { -- Display solution }
        for I := 1 to NumLet do begin
          Write (Letters[UniqueLet[I]], ' = ');
          Writeln (Numbers[UniqueLet[I]]);
        end;  Exit;
        Writeln;  AtLeast1 := True;
      end;
    end;  { -  for N2 }
    if not AtLeast1 then
      Writeln ('NO SOLUTION POSSIBLE');
end.
```

```
{3.8}
program Thr8T83;
{ -- This program will simulate random frog jumps on nine pads. }
uses Crt;
  var
    I, F, Num: Byte;

begin
  Randomize;  ClrScr;
  for I := 1 to 10 do begin
    GotoXY (1, 1);  ClrEol;
    GotoXY (1, 2);  Writeln ('- - - - - - - - -');
    F := 5;
    GotoXY (F*2-1, 1);  Write ('F');  Num := 0;
    repeat
      if Random(2) = 1 then { -- go right }
        Inc(F)
      else  { -- go left }
        Dec(F);
      GotoXY (1, 1);      ClrEol;
      GotoXY (F*2-1, 1);  Write ('F');  Delay (50);
      Inc(Num);
    until (F = 1) or (F = 9);
    GotoXY (I*3, 5);  Write (Num);
  end;
end.
```

```
{3.9}
program Thr9T83;
{ -- This program will allow a user to position a cursor under a
  -- sentence using the L and R keys.  Space bar deletes letter. }
uses Crt;

  var
    I, Col: Byte;
    Sent:   String[80];
    Ch:     Char;

begin
  ClrScr;
  Write ('Enter Sentence: ');  Readln (Sent);  { -- Starts at 17 }
  Col := 17;
  repeat
    GotoXY (Col, 2);
    repeat
      Ch := ReadKey;  Ch := UpCase(Ch);
    until (Ch in ['R', 'L', ' ']) or (Ch = Chr(27));
    if Ch = 'R' then       { -- move cursor to right }
      Inc(Col)
    else if Ch = 'L' then  { -- move cursor to left }
      Dec(Col)
    else if Ch = ' ' then  { -- delete character above cursor }
      Delete (Sent, Col-16, 1);
    GotoXY (17, 1);  Writeln (Sent, ' ');
  until (Ch = Chr(27)) or (Length(Sent) = 1);
end.
```

```
{3.10}
program Thr10T83;
{ -- This program will simulate the movement of a pool ball on a
  -- rectangular pool table.  It moves in a 45 degree angle. }
uses Crt, Graph3;

  var
    W, L, WI, LI, I, X, Y, XDir, YDir: Integer;
    Finished: Boolean;

begin
  Write ('Enter Width, Length: ');  Readln (W, L);
  ClrScr;  GraphMode;
  WI := 10;  LI := 10;
  { -- Draw Pool Table }
  for I := 0 to W do
    Draw (0,I*WI, L*LI,I*WI, 1);
  for I := 0 to L do
    Draw (I*LI,0, I*LI,W*WI, 1);

  X := 0;  Y := W*WI;  XDir := 1;  YDir := -1;
  repeat
    Plot (X, Y, 0);
    X := X + XDir;  Y := Y + YDir;
    Plot (X, Y, 2);  Delay (10);
    if (X = 0) or (X = L*LI) then
      XDir := -1 * XDir;
    if (Y = 0) or (Y = W*WI) then
      YDir := -1 * YDir;

    Finished := True;  GotoXY (1,20);
    if (X = 0) and (Y = 0) then
      Writeln ('LEFT-TOP')
    else if (X = 0) and (Y = W*WI) then
      Writeln ('LEFT-BOTTOM')
    else if (X = L*LI) and (Y = 0) then
      Writeln ('RIGHT-TOP')
    else if (X = L*LI) and (Y = W*WI) then
      Writeln ('RIGHT-BOTTOM')
    else
      Finished := False;
  until Finished;
end.
```

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '84 }
{ -- PASCAL PROGRAM SOLUTIONS }


{1.1}
program One1T84;
{ -- This program produces a table of Fahrenheit for Celcius. }
  var
    C: Integer;

begin
  Writeln ('CELCIUS    FAHRENHEIT');
  C := 50;
  while C <= 200 do begin
    Writeln (C :3, Trunc (1.8 * C + 32 + 0.5) :11);
    C := C + 25;
  end;
end.


{1.2}
program One2T84;
{ -- This program will determine time a person slept in seconds. }
  var
    H1, M1, S1, H2, M2, S2, T: LongInt;

begin
  Write ('WHAT TIME DID YOU GO TO BED (H, M, S) ');
  Readln (H1, M1, S1);
  Write ('WHAT TIME DID YOU GET UP (H, M, S) ');
  Readln (H2, M2, S2);
  T := (11 - H1) * 3600 + (59 - M1) * 60 + (60 - S1);
  Write ('YOU SLEPT FOR ');
  Writeln (T + H2 * 3600 + M2 * 60 + S2, ' SECONDS');
end.


{1.3}
program One3T84;
{ -- This program will display distance/height of a golf ball. }
  var
    T, H, V: Real;

begin
  Writeln (' T    H    V');  T := 0.0;
  while (V > 0) or (T < 1) do begin
    H := 120 * T;  V := 120 * T - 16 * T*T;
    Writeln (T :2:1, '  ', H: 3:0, '  ', V: 3:0);
    T := T + 0.5;
  end;
end.
```

```
{1.4}
program One4T84;
{ -- This program produces table of mice population and food. }
  var
    Y, P, F: Integer;

begin
  Writeln ('NUMBER OF YEARS   POPULATION   FOOD SUPPLY FOR');
  Y := 0;  P := 10;  F := 100;
  Writeln (Y, ' ':16, P :4, F :14);
  while P < F do begin
    Inc(Y);  P := P * 2;  F := F + 40;
    Writeln (Y, ' ':16, P :4, F :14);
  end;
end.
```

```
{1.5}
program One5T84;
{ -- This program will determine time that a savings doubles. }
  var
    N, P, Y: Integer;
    X:       Real;

begin
  Write ('Enter amount, % ');  Readln (N, P);
  X := N;  Y := 0;
  while X < 2 * N do begin
    X := X * (1 + P / 100);  Inc(Y);
  end;
  Writeln (Y, ' YEARS');
end.
```

```
{1.6}
program One6T84;
{ -- This program will determine name at beginning and end. }
  var
    Min, Max, NM: String[10];
    I:            Byte;

begin
  Min := 'ZZZZZZZZZZ';  Max := 'AAAAAAAAAA';
  for I := 1 to 5 do begin
    Write ('Enter name: ');  Readln (NM);
    if NM < Min then Min := NM;
    if NM > Max then Max := NM;
  end;
  Writeln ('NAME CLOSEST TO BEGINNING: ', Min);
  Writeln ('NAME CLOSEST TO END: ', Max);
end.
```

```
{1.7}
program One7T84;
{ -- This program will determine longest run of heads of tosses. }
  var
    N, H, Max, I: Integer;

begin
  Randomize;
  Write ('N: ');  Readln (N);
  H := 0;  Max := 0;
  for I := 1 to N do
    if Random(2) = 1 then Inc(H)
    else
      if H > Max then begin
        Max := H;  H := 0;  end
      else
        H := 0;
  If H > Max then Max := H;
  Writeln (Max, ' CONSECUTIVE HEADS');
end.
```

```
{1.8}
program One8T84;
{ -- This program will display numbers with 7s zapped. }
  var
    I, T, O: Byte;

begin
  for I := 1 to 100 do begin
    T := I div 10;  O := I - T * 10;
    if ((T = 7) or (O = 7)) and (I mod 7 = 0) then
      Write ('ZAPZAP' :16)
    else if (T = 7) or (O = 7) then
      Write ('ZAP': 16)
    else
      Write (I :16);
  end;
  Writeln;
end.
```

```
{1.9}
program One9T84;
{ -- This program will print the # of double letters. }
  var
    C, LastC: Char;
    A:        String[60];
    D, I:     Byte;

begin
  Write ('Enter text: ');  Readln (A);  D := 0;
  for I := 1 to Length(A) do begin
    C := A[I];
    if C = LastC then Inc(D);
    LastC := C;
  end;
  Writeln (D);
end.
```

```
{1.10}
program One10T84;
{ -- This program will display sevens multiplication facts. }
  var
    I, Ans, W: Byte;

begin
  for I := 0 to 9 do begin
    W := 0;
    repeat
      Write (I, ' X 7 = ');  Readln (Ans);
      if Ans <> I * 7 then
        if W = 0 then W := 1 else begin
           Writeln (I * 7);
           W := 2;
         end;
    until (I * 7 = Ans) or (W = 2);
  end;
end.
```

```
{2.1}
program Two1T84;
{ -- This program will print number of vowels in text. }
  var
    A:    String[60];
    C:    Char;
    I, V: Byte;

begin
  Write ('Enter text: ');  Readln (A);
  for I := 1 to Length (A) do begin
    C := A[I];
    if C in ['A', 'E', 'I', 'O', 'U'] then
      Inc(V);
  end;
  Writeln (V, ' VOWELS');
end.


{2.2}
program Two2T84;
{ -- This program sorts rational numbers in increasing order. }
  var
    N, M, I, J, S: Integer;
    Nst, Mst, Xst: String[7];
    X: Real;
    V: Array [1..9] of Real;
    A: Array [1..9] of String[7];

begin
  Write ('Enter N, M: ');  Readln (N, M);  S := 0;
  while (N > 0) and (M > 0) do begin
    Inc(S);
    Str (N, Nst);  Str (M, Mst);
    A[S] := Nst + '/' + Mst;  V[S] := N / M;
    Write ('Enter N, M: ');  Readln (N, M);
  end;
  for I := 1 to S-1 do
    for J := I+1 to S do
      if V[I] > V[J] then begin
        X := V[I];    V[I] := V[J];  V[J] := X;
        Xst := A[I];  A[I] := A[J];  A[J] := Xst;
      end;
  for I := 1 to S do Writeln (A[I]);
end.
```

```
{2.3}
program Two3T84;
{ -- This program displays #s that sum of cubes of digits= #. }
  var
    I, J, K, Num: Integer;

begin
  for I := 1 to 9 do
    for J := 0 to 9 do
      for K := 0 to 9 do begin
        Num := I*100 + J*10 + K;
        if Num = I*I*I + J*J*J + K*K*K then Writeln (Num);
      end;
end.
```

```
{2.4}
program Two4T84;
{ -- This program will print a triangle of #s by an algorithm. }
  var
    N, J, I, X: Integer;

begin
  Write ('Enter # of rows: ');  Readln (N);
  for I := 1 to N do begin
    Write (' ': N-I+1);
    for J := I to 2*I - 1 do
      Write (J mod 10);
    for J := 2*I - 2 downto I do
      Write (J mod 10);
    Writeln;
  end;
end.
```

```
{2.5}
program Two5T84;
{ -- This program will display a page of multiplication drills. }
uses Crt;
  var
    I, H, V, X, Y: Byte;

begin
  Randomize;  ClrScr;
  Writeln ('     MULTIPLICATION DRILL');
  for I := 1 to 6 do begin
    H := (I - 1) div 3;  V := I - H * 3;  H := H * 20 + 1;
    X := Random(90) + 10;  Y := Random(9) + 1;
    GotoXY (H, V*5);  Write (I, '.  ', X);
    GotoXY (H, V*5+1);  Write ('   X ', Y);
    GotoXY (H, V*5+2);  Write ('   ----');
  end;
end.
```

```
{2.6}
program Two6T84;
{ -- This program will simulate throwing darts. }
  var
    N, X, Y, I, J, S: Byte;
    A: Array [1..5, 1..5] of Byte;

begin
  Randomize;  Write ('Enter N: ');  Readln (N);  S := 0;
  for I := 1 to 5 do
    for J := 1 to 5 do
      A[I, J] := 0;
  for I := 1 to N do begin
    X := Random(5) + 1;  Y := Random(5) + 1;  A[X, Y] := 1;
  end;
  for I := 1 to 5 do begin
    for J := 1 to 5 do
      if A[I, J] = 1 then begin
        Write ('* ');  Inc(S);  end
      else
        Write ('. ');
    Writeln;
  end;
  Writeln ('NUMBER OF THROWS = ', N);
  Writeln ('NUMBER OF SQUARES HIT = ', S);
end.
```

```
{2.7}
program Two7T84;
{ -- This program will determine if text is palindrome. }
  var
    A, S: String[80];
    L, I: Byte;
    C:    Char;

begin
  Write ('Enter text: ');  Readln (A);
  S := '';
  for I := 1 to Length(A) do begin
    C := A[I];
    if (C >= 'A') and (C <= 'Z') then S := S + C;
  end;
  L := Length(S);
  for I := 1 to L div 2 do
    if Copy(S, I, 1) <> Copy(S, L - I + 1, 1) then begin
      Writeln ('NOT PALINDROME');  Exit;
    end;
  Writeln ('PALINDROME');
end.
```

```
{2.8}
program Two8T84;
{ -- This program will display the frequency of letters. }
  var
    A: String[60];
    B: Array[1..26] of Byte;
    L, I, X, T:        Byte;
    C:                 Char;

begin
  Write ('Enter sentence: ');  Readln (A);
  L := Length(A);  T := 0;
  for I := 1 to 26 do B[I] := 0;
  for I := 1 to L do begin
    C := A[I];
    if C in ['A' .. 'Z'] then begin
      X := Ord(C) - Ord('A') + 1;  Inc(B[X]);  Inc(T);
    end;
  end;
  Writeln ('LETTER   FREQUENCY   PERCENT');
  for I := 1 to 26 do
    if B[I] > 0 then begin
      Write (Chr(I + 64), ' ':8, B[I], ' ':11);
      Writeln (Round (B[I] / T * 100));
    end;
    Writeln ('TOTAL ', T);
end.
```

```
{2.9}
program Two9T84;
{ -- This program will print the longest word in sentence. }
  var
    A, W, Max: String[80];
    I, L:      Byte;
    C:         Char;

begin
  Write ('Enter sentence: ');  Readln (A);  A := A + ' ';
  L := Length (A);  Max := '';  W := '';
  for I := 1 to L do begin
    C := A[I];
    if C <> ' ' then
      W := W + C
    else begin
      if Length(W) > Length(Max) then Max := W;
      W := '';
    end;
  end;
  Writeln (Max);
end.
```

```
{2.10}
program Two10T84;
{ -- This program will play rock, scissors, and paper. }
  var
    A:          Char;
    X, T, L, W: Byte;

begin
  Randomize;
  Write ('Enter R, S, P, or Q: ');  Readln (A);
  W := 0;  L := 0;  T := 0;
  while A <> 'Q' do begin
    X := Random (3);
    if      (X = 0) and (A = 'R') then begin
      Inc(T);  Writeln ('TIE');  end
    else if (X = 1) and (A = 'S') then begin
      Inc(T);  Writeln ('TIE');  end
    else if (X = 2) and (A = 'P') then begin
      Inc(T);  Writeln ('TIE');  end
    else if (X = 0) and (A = 'P') then begin
      Inc(W);  Writeln ('YOU WIN');  end
    else if (X = 1) and (A = 'R') then begin
      Inc(W);  Writeln ('YOU WIN');  end
    else if (X = 2) and (A = 'S') then begin
      Inc(W);  Writeln ('YOU WIN');  end
    else begin
      Inc(L);  Writeln ('I WIN');
    end;
    Write ('Enter R, S, P, or Q: ');  Readln (A);
  end;
  Writeln (T, ' TIES');
  Writeln (W, ' WINS (YOURS)');
  Writeln (L, ' LOSSES (MINE)');
end.
```

```
{3.1}
program Thr1T84;
{ -- This program will display a random trail of asterisks. }
{ -- However, the program description is poorly worded and
     ambiguous.  The judging criteria is also poorly described. }
uses Crt;
  var
    A: Array [1..24, 1..80] of Byte;
    I, J, V, H, X, Y:          Byte;
    Ch:                        Char;
    SameRun:                   Boolean;

begin
  Randomize;
  repeat
    ClrScr;
    for I := 1 to 24 do
      for J := 1 to 80 do A[I,J] := 0;
    V := 12;  H := 40;  A[V, H] := 1;
    GotoXY (H, V);  Write ('S');  SameRun := True;

    while SameRun do begin
    repeat
      X := Random(4)
    until (X - 2 <> Y) and (Y - 2 <> X);
    if X = 0 then Dec(H);
    if X = 2 then Inc(H);
    if X = 1 then Dec(V);
    if X = 3 then Inc(V);
    if (A[V,H] = 1) or (V = 0) or (V = 23) or (H = 0) or (H = 80)
    then begin
      GotoXY (1, 22);
      Write ('THE MAXIMUM DISTANCE FROM START = ');
      Writeln (Abs(40 - H) + Abs(12 - V));
      Ch := ReadKey;  SameRun := False;
    end
    else begin
      A[V, H] := 1;  GotoXY (H, V);  Write ('*');  Y := X;
    end;
    end;  { -- while }
  until Ch = 'Q';
end.
```

```
{3.2}
program Thr2T84;
{ -- This program will decode a message with frequent letters. }
  const
    B: String[12] = 'ETAOINSHRDLU';
  var
    Ast, Bst: Array [0..32] of Char;
    A:        Array [1..32] of Byte;
    Mes:      String[32];
    I, J, K,
    L, S, G:  Byte;

begin
  Write ('Message: ');  Readln (Mes);  L := Length(Mes);
  for I := 1 to L do begin
     Ast[I] := Mes[I];  A[I] := 0;
  end;
  Ast[0] := ' ';  G := 0;  S := 0;
  for I := 1 to L do begin
    K := 0;
    while (Ast[K] <> Ast[I]) and (K <= I-1) do Inc(K);
    if K = I then begin  { -- Found 1st occurence of letter }
      for J := I to L do
        if Ast[I] = Ast[J] then Inc(A[I]);
      if A[I] > G then G := A[I];
    end;
  end;
  { -- Replace letters in message }
  for I := G downto 1 do begin
    J := 1;
    while (A[J] <> I) and (J <= L) do Inc(J);
    if J <= L then begin
      Inc(S);
      for K := J to L do
        if Ast[K] = Ast[J] then Bst[K] := B[S];
    end;
  end;
  for I := 1 to L do Write (Bst[I]);
  Writeln;
end.
```

```
{3.3}
program Thr3T84;
{ -- This program will produce the digital product root. }
  var
    I:         Byte;
    Nst, N, X: LongInt;

begin
  Write ('ORIGINAL VALUE (1 TO 7 DIGITS): ');  Readln (Nst);
  Writeln (Nst);
  while Nst > 9 do begin
    N := 1;
    for I := 1 to trunc(ln(Nst) / ln(10)) + 1 do begin
      X := Nst - (Nst div 10) * 10;
      if X > 0 then N := N * X;
      Nst := Nst div 10;
    end;
    Writeln (N);  Nst := N;
  end;
end.
```

```
{3.4}
program Thr4T84;
{ -- This program will display twin primes. }
  var
    N, I, J, T: Integer;
    Prime:      Boolean;

begin
  Write ('Enter N: ');  Readln (N);
  Writeln ('TWIN PRIMES NOT GREATER THAN ', N);
  for I := 3 to N - 2 do begin
    J := 2;  Prime := True;
    while (J <= Trunc(Sqrt(I))) and Prime do begin
      If I mod J = 0 then Prime := False;
      Inc(J);
    end;
    if Prime then begin
      T := I + 2;
      J := 2;
      while (J <= Trunc(Sqrt(T))) and Prime do begin
        if T mod J = 0 then Prime := False;
        Inc(J);
      end;
      if Prime then Writeln (I, '  ', T);
    end;
  end;
end.
```

```
{3.5}
program Thr5T84;
{ -- This program will print subsets of m people. }
  var
    A:   Array [1..26] of Byte;
    Ast: Array [1..26] of Char;
    I, M, L, N, S:       Byte;

begin
  Write ('INPUT NUMBER, CAPACITY: ');  Readln (L, M);
  for I := 1 to M do   A[I] := M - I + 1;
  for I := 1 to L do Ast[I] := Chr(64 + I);
  N := 1;  Dec(A[1]);  S := 0;
  while N <= M do begin
    Inc(A[N]);
    if N > 1 then
      for I := N-1 downto 1 do A[I] := A[I+1] + 1;
    if A[N] <= L - N + 1 then begin
      for I := M downto 1 do Write (Ast[A[I]]);
      Write(' ': 16 - M);
      Inc(S);  N := 0;
    end;
    Inc(N);
  end;
  Writeln;
  Writeln ('THERE ARE ', S, ' SUBSETS');
end.
```

```
{3.6}
program Thr6T84;
{ -- This program will display histogram of letter frequency. }
uses Crt;
  const
    B: Array [1..5] of String[50] =
      ('THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.',
       'THIS IS AN EXAMPLE OF HOW',
       'TO TEST YOUR HISTOGRAM PROGRAM.  YOU',
       'CAN USE THIS EXAMPLE.',
       '*END*');
  var
    A:          Array [1..26] of Byte;
    I, J, X, G: Byte;
```

```
begin
  ClrScr;
  for I := 1 to 26 do A[I] := 0;
  J := 1;  G := 0;
  while B[J] <> '*END*' do begin
    for I := 1 to Length(B[J]) do begin
      X := Ord(B[J, I]) - Ord('A') + 1;
      if (X >= 1) and (X <= 26) then
        Inc(A[X]);
      if A[X] > G then G := A[X];
    end;
    Inc(J);
  end;
  for I := G downto 1 do begin
    for J := 1 to 26 do
      if A[J] >= I then begin
        GotoXY (J, G - I + 1);  Write ('*');
      end;
    Writeln;
  end;
  for I := Ord('A') to Ord('Z') do Write (Chr(I));
  Writeln;
end.
```

```
{3.7}
program Thr7T84;
{ -- This program will display a repeating decimal. }
  var
    Re: Array [1..100] of Integer;
    N, D, X, I, J, R:     Integer;
    A, Xst:               String[100];

begin
  A := '';  I := 0;
  Write ('Enter N, D: ');  Readln (N, D);
  Write (N, '/', D, ' = ');  X := N div D;
  if X > 0 then Write (X);
  Write ('.');
  repeat
    Inc(I);  R := N - D * X;
    if R = 0 then begin
      Writeln (A);  Exit;
    end;
    Re[I] := R;  N := R * 10;  X := N div D;
    { -- Display decimal if remainder repeats itself }
    for J := 1 to I - 1 do
      if Re[J] = R then begin
        Write   (Copy(A, 1, J-1), '(');
        Writeln (Copy(A, J, I-J), ')');  Exit;
      end;
    Str (X, Xst);
    A := A + Xst;
  until R = 0;
end.
```

```
{3.8}
program Thr8T84;
{ -- This program will print # of round numbers less than N. }
  var
    I, J, K, L, M, N, S, T, X, Pow: Integer;

begin
  Write ('INPUT NUMBER: ');  Readln (N);  T := 0;
  for I := 2 to N do begin
    M := I;  S := 0;  K := Trunc(Ln(M) / Ln(2) + 0.01);
    for J := K downto 0 do begin
      Pow := 1;
      for L := 1 to J do Pow := Pow * 2;
      X := M div Pow;
      S := S + X;  M := M - X * Pow;
    end;
    if S + S = K + 1 then Inc(T);
  end;
  Write   ('THERE ARE ', T);
  Writeln (' ROUND NUMBERS LESS THAN OR EQUAL TO ', N);
end.
```

```
{3.9}
program Thr9T84;
{ -- This program will provide automated price increases. }
  const
    A: Array [1..3] of String[50] =
      ('THE CURRENT COST OF BUCKLES IS',
       '3 FOR $2.50, OR $10.00 A DOZEN.',
       '*END*');
  var
    I, J, K, L, X, Per, Code: Integer;
    Xst:  Char;
    P, T: Real;

begin
  Write ('Enter %: ');  Readln (P);  P := P / 100;
  K := 1;
  while A[K] <> '*END*' do begin
    L := Length (A[K]);  I := 0;
    repeat
      Per := 0;
      while (I < L) and (Xst <> '$') do begin
        Inc(I);  Xst := A[K , I];  Write (Xst);
      end;
      if Xst <> '$' then Writeln
      else begin
        J := I;  X := 50;
        while (J < L) and ((Xst = '.') or ((X > 47) and (X < 58)))
        and (Per < 2) do begin
          Inc(J);  Xst := A[K , J];  X := Ord(Xst);
          if Xst = '.' then Inc(Per);
        end;
        Val (Copy(A[K], I+1, J-I-1), T, Code);
        T := T + T * P;  T := Round(T * 100) / 100;
        Write (T: 4:2);
        I := J - 1;
      end;
    until I >= L;
    Inc(K);
  end;
end.
```

```
{3.10}
program Thr10T84;
{ -- This program will simulate tennis sets between 2 players. }
  var
    N, P, A, B, AG, BG, W, L: Integer;

begin
  Randomize;  A := 0;  B := 0;  AG := 0;  BG := 0; W := 0; L := 0;
  Write ('NUMBER OF SETS = ');  Readln (N);
  Write ('% CHANCE A WINS A POINT= ');  Readln (P);
  repeat
    if Random(100) < P then Inc(A) else Inc(B);
    if (A > 3) and (A > B + 1) then begin
      Write ('A');  Inc(AG);  A := 0;  B := 0;
    end;
    if (B > 3) and (B > A + 1) then begin
      Write ('B');  Inc(BG);  A := 0;  B := 0;
    end;
    if (AG > 5) and (AG > BG + 1) then begin
      Writeln ('  (A)');  Inc(W);  AG := 0;  BG := 0;
    end;
    if (BG > 5) and (BG > AG + 1) then begin
      Writeln ('  (B)');  Inc(L);  AG := 0;  BG := 0;
    end;
  until W + L = N;
  if W > L then
    Writeln ('PLAYER ''A'' WON ', W, ' SETS OUT OF ', N)
  else
    Writeln ('PLAYER ''B'' WON ', L, ' SETS OUT OF ', N);
end.
```