HEWLETT-PACKARD

KEYBOARD

VOL. 7 NO. 4

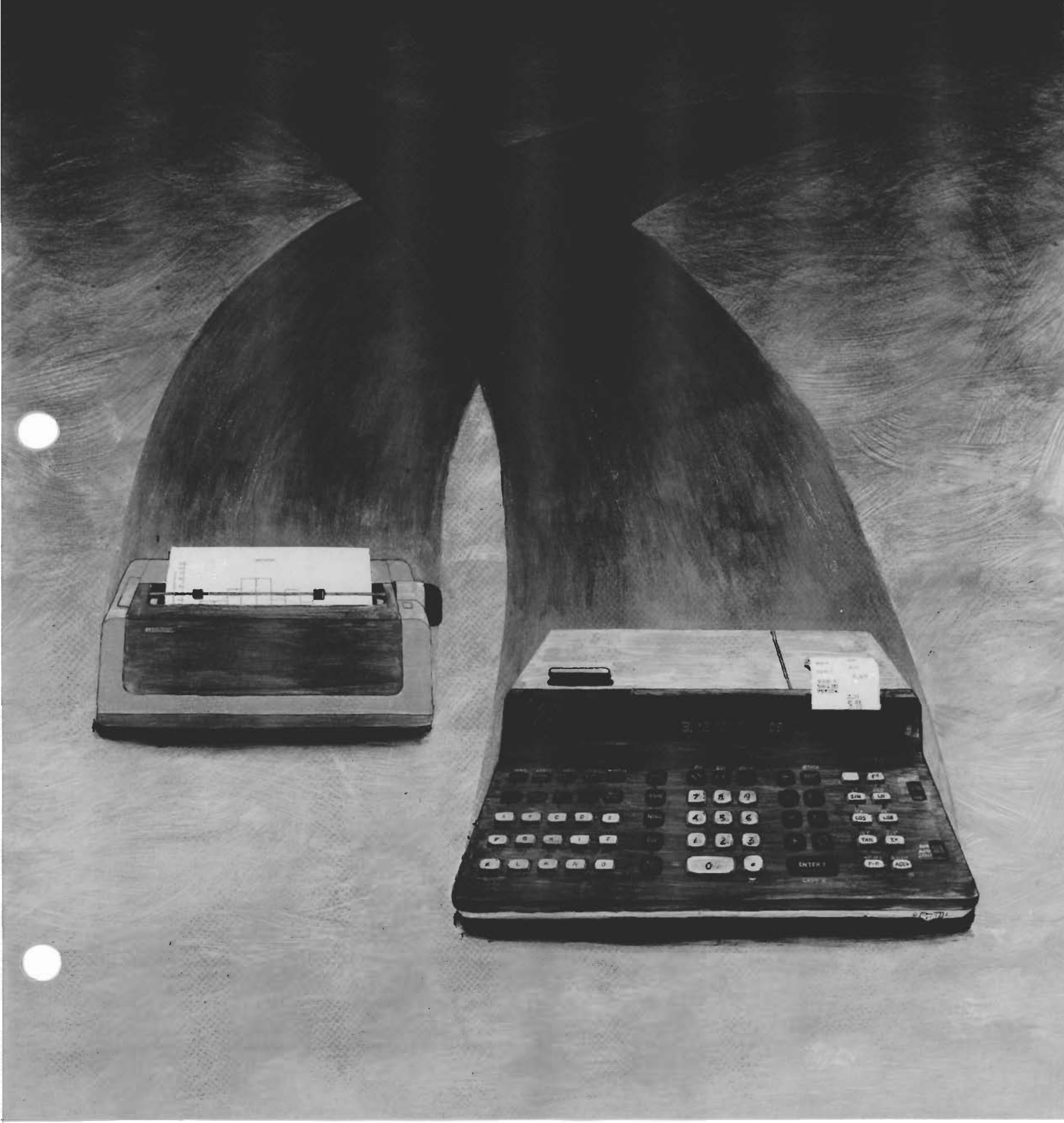


TABLE OF CONTENTS

OVERVIEW

This is our last issue for 1975, and, by the time you receive it, I'll have been editor for a full year. Although some minor changes have been made, I've tried to retain the basic philosophy of *KEYBOARD*. (Hopefully, I've enhanced it.) And this philosophy is simple — maintain a three-way communications link: reader to reader, reader to Hewlett-Packard, Hewlett-Packard to reader. Sharing information and experience with other users is the foremost link in the *KEYBOARD* communications system. We value the inputs you give us and hope we return in kind with a helpful, informative magazine. Special appreciation goes to all those who have contributed articles and programming tips.

In 1976 we'll be working toward making *KEYBOARD* even better. We hope you'll continue to be pleased with our efforts.

Nancy Sorensen

Forum

We are interested in obtaining programs written for use on the 9820A pertaining to the field of soil mechanics and geotechnical engineering.

Particular applications of interest would be slope stability analysis, soil laboratory analysis, surveying, etc.

We have a few programs that we could exchange (curve-tracing for various types of inclinometers and deflectometers, curve-tracing for static penetrometers).

J. Y. Chagnon, Director
Geotechnical Service
Department of Natural Resources
1600, Blvd. Entente
Quebec
GIS 4N6
Canada

We enthusiastically endorse the creation of your "Forum" column. We have been using the 9830A for almost two years in our structural engineering work.

We would be interested in hearing from other structural engineers and learning what programs they have written. Our office has written a group of structural engineering programs as well as several business programs suitable for a consulting office.

Included in the first group are such programs as analysis of continuous beams, general plane frames, cantilever retaining walls, prestressed beams, laterally loaded piles and composite beams. The business programs include payroll, bookkeeping and accounts receivable.

Interested parties are welcome to contact us.

Robert Stoller
Zeiler and Gray, Inc.
2727 Bryant Street
Denver, Colorado 80211
(303) 455-3322

APPLICATIONS INFORMATION
FOR HEWLETT-PACKARD CALCULATORS
PUBLISHED AT P.O. BOX 301,
LOVELAND, COLORADO 80537

Editor: Nancy Sorensen — Artist/Illustrator: H. V. Andersen

Field Editors: **ASIA**--Jaroslav Byma, Hewlett-Packard Intercontinental, 3200 Hillview Avenue, Palo Alto, California 94304; **AUSTRALASIA**--Bill Thomas, Hewlett-Packard Australia Pty. Ltd., 31-51 Joseph Street, Blackburn, 3130 Victoria, Australia; **BELGIUM**--Luc Desmedt, Hewlett-Packard Benelux, Avenue du Col-Vert, 1, Groenkraaglaan, B-1170 Brussels, Belgium; **CANADA**--Larry Gillard, Hewlett-Packard Canada Ltd., 6877 Goreway Drive, Mississauga, Ontario L4V 1L9; **EUROPEAN REGIONAL EDITOR**--Ed Hop, Hewlett-Packard GmbH, Herrenbergerstrasse 110, 7030 Böblingen, Germany; **EASTERN AREA, EUROPE**--Werner Hascher, Hewlett-Packard Ges.m.b.H., Handelskai 52/3, A-1205 Vienna, Austria; **ENGLAND**--Dick Adaway, Hewlett-Packard Ltd., King Street Lane, Winnersh, Wokingham, England; **FRANCE**--Elisabeth Caloyannis, Hewlett-Packard France, Quartier de Courtaboeuf, Boîte Postale No. 6, F-91401 Orsay, France; **GERMANY**--Rudi Lamprecht, Hewlett-Packard GmbH, Berner Strasse 117, D-6000 Frankfurt 56, Germany; **HOLLAND**--Jaap Vegter, Hewlett-Packard Benelux N.V., Van Heuven Goedhartlaan 121, P.O. Box 667, NL-1134 Amstelveen, Holland; **ITALY**--Elio Doralio, Hewlett-Packard Italiana Spa, Via Amerigo Vespucci 2, I-20124, Milano, Italy; **JAPAN**--Akira Saito, Yokogawa-Hewlett-Packard Ltd., 59-1, Yoyogi 1-chome, Shibuya-ku, Tokyo 151; **LATIN AMERICA**--Ed Jaramillo, Hewlett-Packard Intercontinental, 3200 Hillview Avenue, Palo Alto, California 94304; **MIDDLE EAST**--Philip Pote, Hewlett-Packard S.A., Mediterranean and Middle East Operations, 35, Kolokotroni Street, Platia Kefallariou, GR-Kifissia-Athens, Greece; **SCANDINAVIA**--Per Styme, Hewlett-Packard Sverige AB, Enighetsvägen 3, Fack, S-161 20 Bromma 20, Sweden; **SOUTH AFRICA**--Denis du Buisson, Hewlett-Packard South Africa (Pty.) Ltd., 30 de Beer Street, Braamfontein; **SPAIN**--José L. Barra, Hewlett-Packard Española S.A., Jerez 3, E — Madrid 16, Spain; **SWITZERLAND**--Heinz Neiger, Hewlett-Packard Schweiz AG, Zürcherstrasse 20, P.O. Box 64, CH-8952 Schlieren, Zürich, Switzerland.

[illegible]

Hewlett-Packard has announced the first printer designed specifically to interface with the 9800 Series programmable calculators. The 9871A Printer is a 30-character-per-second, full-character impact printer using interchangeable print discs. Each print disc contains 96 printing characters. The printer is fully self-contained and requires only a power cord and calculator interface cable to replace existing printing devices on any of the 9800 Series programmable calculators. Additional hardware features include a fixed carriage accommodating paper up to 15 inches wide. Print width can be up to 13.2 inches, which, at the normal print spacing of 10 characters per inch, gives 132 columns. Bidirectional print disc carrier and platen motion simplifies two additional features — form filling and plotting. The optional form-feed mechanism is recommended for Z-fold, continuous-feed, or multiple-part paper (the 9871A is capable of handling up to 6-part paper). A 10-position print intensity switch is located on the rear panel to insure good print quality when using different thicknesses of paper.

One of the unique features of the 9871A is the built-in, self-test capability. Located on the rear panel is a test pushbutton switch which, when depressed, will cause the 9871A to perform a built-in diagnostic. The diagnostic causes the printer to print out a line of type, advance the platen, and check the internal memory of the 9871A Printer. If an error is detected, an audible beep emits from the printer to alert the operator that something is wrong.

The normal printing functions, fully controllable from the calculator, include space and backspace, carrier return, line feed, reverse line feed, and variable view advance delay (after completing a line of text, the platen reverses to allow the operator to view that line).

Other programmable printer operations include:

Horizontal Tabulation
- Set horizontal tab
- Clear horizontal tab
- Clear all horizontal tabs
- Tab right
- Tab left

Vertical Tabulation
- Set vertical tab
- Clear vertical tab
- Clear all vertical tabs
- Tab down
- Tab up

Form and Margin Control
- Set top of form
- Set form length
- Form feed
- Set left margin
- Set text width
- Set text length (initiates automatic form feed when text length is exceeded)

Programmable plotting and spacing functions are:

Plotting Control
- Absolute plot (resolution 1/120 in x 1/96 in)
- Relative plot (resolution 1/120 in x 1/96 in)
- Absolute plot with points filled between end points
- Relative plot with points filled between end points
- Set origin for absolute plotting

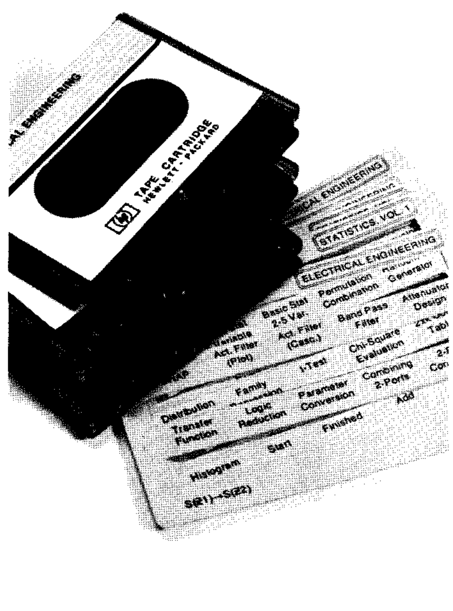
Spacing Control
- Variable horizontal spacing (resolution 1/120 in)
- Variable vertical spacing (resolution 1/96 in)
- Proportional spacing

The 9871A Data Sheet, 5952-9000 (09), is available upon request. For literature, information, or a demonstration, please contact your local Hewlett-Packard sales office or check the reply card in this issue of *KEYBOARD*.

9815A

Hewlett-Packard has announced a new programmable desktop calculator, the 9815A. This newest member of scientifically oriented calculators uses the same RPN language as the HP pocket calculators and is designed for broad-based applications among a wide range of users who work with statistics and who make computations particular to their professions, as in scientific, engineering, research, and industrial fields.

The resourcefulness of the RPN language has been demonstrated by the 9100A/B, the 9810A, and the series of HP pocket calculators. The 9815 further enhances this language with multiple flags, both computed "go to" and computed label search, FOR-NEXT loop capacity, compilation of multiple keystroke operations, and a greater choice of branching techniques.



Human engineering is an important aspect of the design of the 9815. For instance, it has a very simple keyboard. Anyone who has used one of our pocket calculators will immediately recognize most of the keys. Editing features are improved — insert and delete functions that automatically update all branching instructions, error messages in English, complete alpha listings, entry flag, error detection flags, and simpler UDF keys.

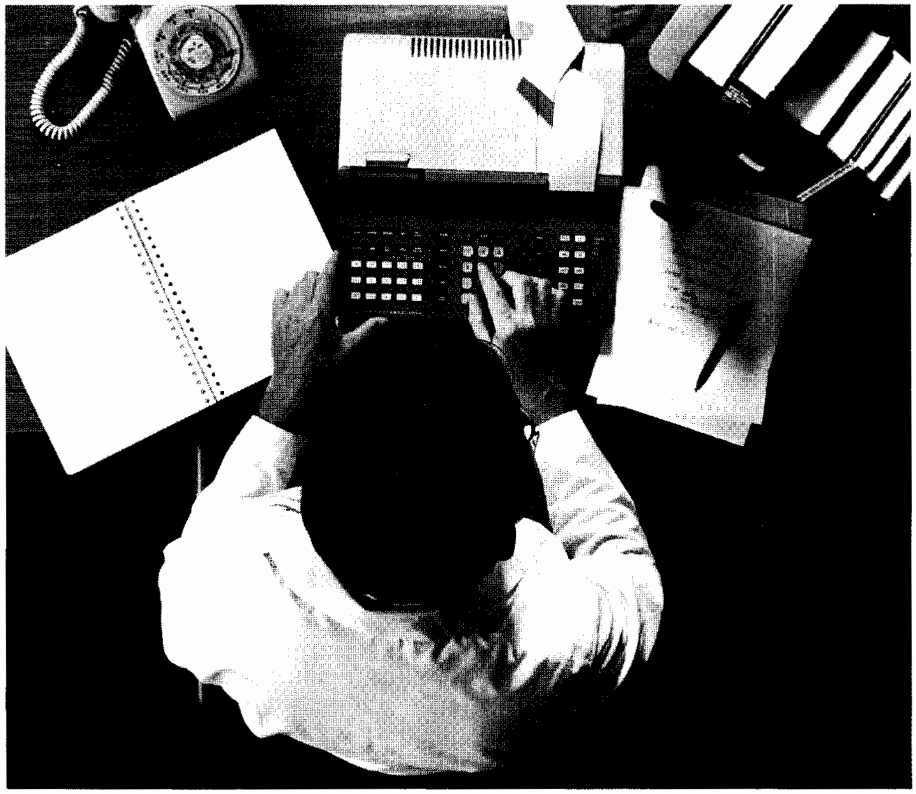
With the revolutionary, high-speed bidirectional tape cartridge, the HP 9815 offers true, state-of-the-art performance. The cartridge holds 96,384 program steps, or 12,048 data registers, or any combination between these two limits. The 9815 is thus able to handle much larger problems than other machines in its price range. Bidirectional search speed of 1524 mm/sec (60 in/sec) and read/write speed of 254 mm/sec (10 in/sec) gives you nearly instantaneous access to data and programs. A 500-step program is loaded and execution begun in about 0.6 second, and a 2000-step program in under 2 seconds.

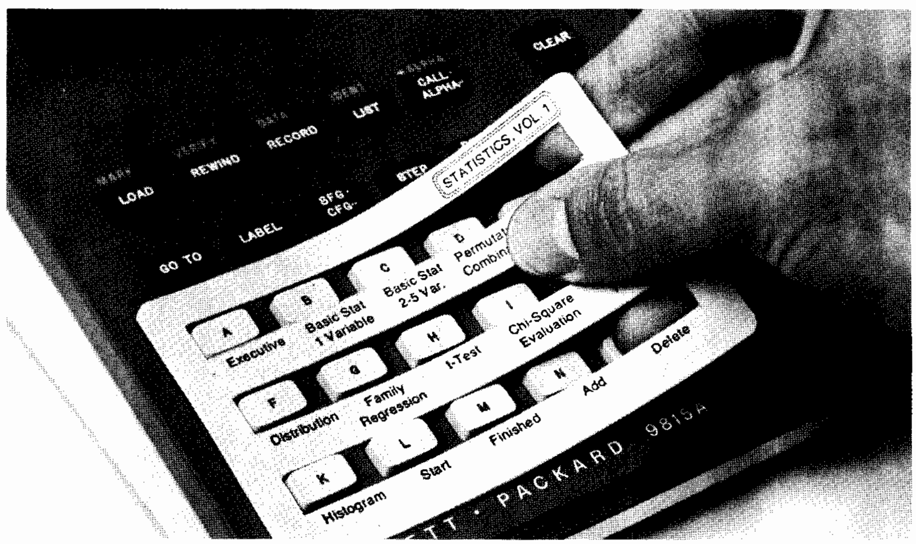
The 9815 contains a built-in thermal printer with alphanumerics and mathematic and trigonometric functions. The standard 9815 has 472 program steps and 10 data registers. Memory is expanded to 2008 program steps in the 9815 Option 001. The keyboard includes 15 special function keys, a 10-key numeric pad, program language and control keys, editing keys, and 28 scientific functions.

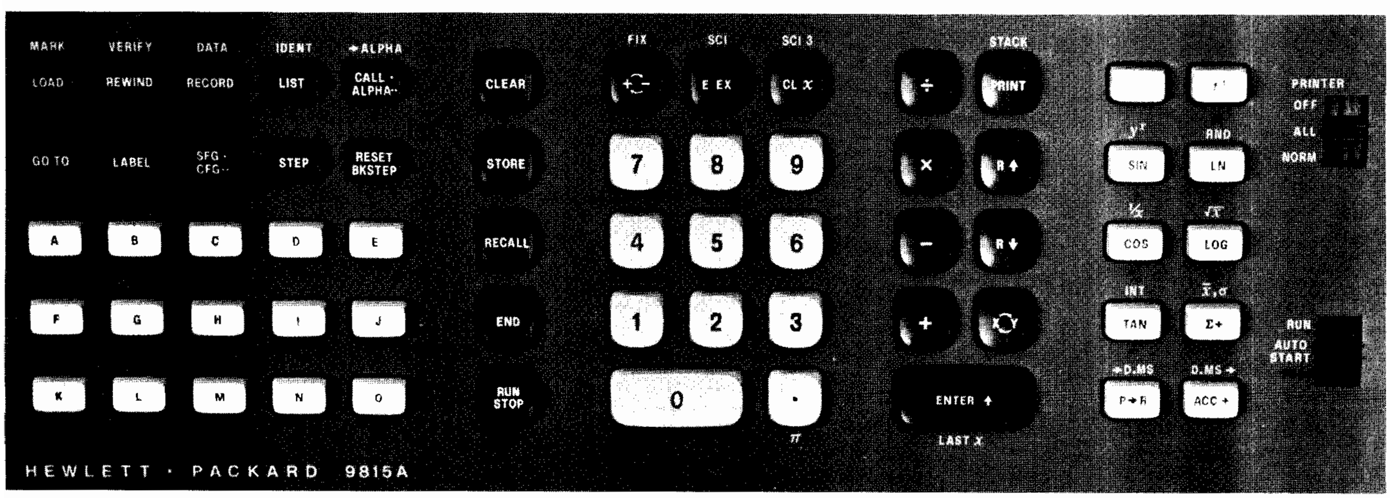
Users interested in interfacing to experimental or process equipment will also find the 9815 a well-designed, versatile machine. Basic I/O speed of the 9815 Option 002 is much faster than I/O speed of our other 9800 Series Calculators. Such features as burst mode, binary operations, buffered input and output, and programmable delimiters are built in. Logic sense and the polarity of the flag and control lines are changeable under program control. The AUTO START feature allows the machine to recover in the event of power failure or interruption.

The 9815 Option 002 has two I/O channels with plug-to-plug compatibility with many HP 9800 Series peripherals and a wide range of digital voltmeters, counters, and other instruments. Three general types of interface cards are available: general 8-bit I/O, BCD input, and HP-IB. Up to 14 HP-IB instruments can be interconnected to a single HP-IB interface card, and a 15th instrument can be interfaced via the second channel available.

A brochure, "HP 9815A: A Four-Dimensional Machine," 5952-8998 (09), and the 9815A Data Sheet, 5952-8999 (09), are available upon request. For literature, information, or a demonstration, please contact your local Hewlett-Packard sales office or check the reply card in this issue of *KEYBOARD*.







Generalized Linear Least-Squares Fitting

by Dr. Ove W. Dietrich and Mr. Ole S. Rathmann

Thanks to the development of computers and automation, scientists of today can handle problems which their predecessors could not even think of. Apart from the obvious advances in science in general, this development has also had an impact on many of the scientists themselves, theorists as well as experimentalists. The vast amounts of data, which can be stored and handled by the computers, have turned many modern scientists into data analysts. They must know how to extract meaningful results and conclusions from the mass of numbers they feed into the computer memory. The technique of data reduction varies from one problem to the next, but there are several standard methods that most scientists use at some time or other.

In this article we describe such a standard procedure that is useful in many contexts. The "generalized linear least-squares fit" has long been known, but it is probably new to some *KEYBOARD* readers. Hopefully, it is good news to many readers that this fairly lengthy procedure can actually be executed on a small desk calculator like the HP 9830A with 2k memory. In the following we shall briefly describe the fitting procedure and give reasons why and how it is useful.

The simplest data reduction problem is, undoubtedly, finding the average value of a set of numbers. Although this must be familiar to all our readers, we shall start by defining our aims in this simple context. Let us imagine that we have measured the length of a stick four times and got slightly different results each time. We would then like to find the best estimate of the length. If the measurements gave the results of 1.015 m, 1.018 m, 1.016 m, and 1.017 m, the best estimate would, of course, be the average value, 1.0165 m. However, there is a little more information in these numbers that we would like to extract. From the scatter of the individual measurements around the average value we can estimate the uncertainty or standard error of the average value, if we assume that the measurements deviate according to the so-called normal distribution. Denoting the individual measurements by y_i with $i=1$ up to N and the average value by $\bar{y}$, the uncertainty of the average value is

$$S_{\bar{y}} = \left(\frac{\Sigma\,(y_i - \bar{y})^2}{n(n-1)}\right)^{1/2}.$$

Thus for the numbers above, the average value including the uncertainty will be

$$\bar{y} = 1.0165 \pm 0.0006.$$

The statistical meaning of this uncertainty is such that if we repeat the set of measurements many times, we would expect that approximately 2/3 of the average values for each set would fall within the range

$$\bar{y} - S_{\bar{y}} \text{ and } \bar{y} + S_{\bar{y}}.$$

Let us complicate the example somewhat and assume that we have used two different measuring devices to obtain the lengths. The first three measurements were made with a simple ruler with a precision of 0.001 m, whereas the last measurement was made with a slide gage with a precision of 0.0002 m. Now we have to assign different weights to the measurements. The weight w_i is defined as the inverse square of the precision σ_i, i.e.

$$w_i = \sigma_i^{-2}.$$

The average value is then

$$\bar{y} = \frac{\Sigma w_i\, y_i}{\Sigma w_i},$$

and the uncertainty of $\bar{y}$ is

$$S_{\bar{y}} = \left(\frac{\Sigma w_i\,(y_i - \bar{y})^2}{(n-1)\,\Sigma w_i}\right)^{1/2}.$$

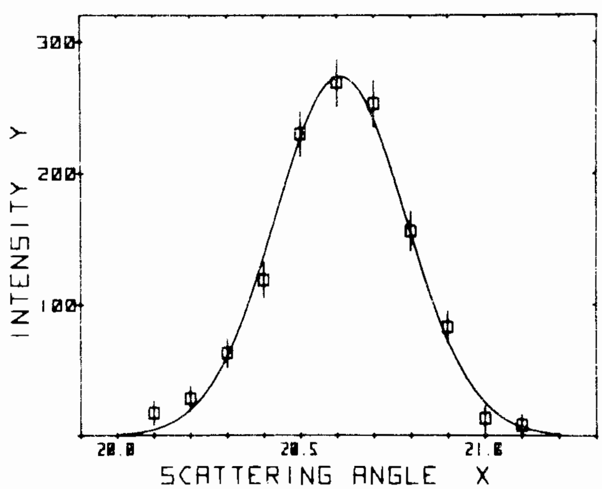
For this case our best estimate of the average value is calculated to be

$$\bar{y} = 1.0169 \pm 0.0003.$$

To relate the average value problem to least-squares fittings, we notice that the average value we have determined above is the number giving the least sum of the weighted mean squared deviations,

$$X^2 = \frac{1}{(n-1)}\,\Sigma w_i\,(y_i - \bar{y})^2\,.$$

The same criterion is used in more complex fitting problems too. In the example above we had only one variable, y_i, and one parameter to be determined, the average value $\bar{y}$. However, we often have two variables, an independent and a dependent variable, and possibly more than one parameter to be determined from a set of measurements. Let us illustrate a more complicated case by an example:



In the diagram we show some data measured with a spectrometer. In this instrument particles or rays can be scattered in a sample and the scattered intensity measured as a function of scattering angle. The present data were obtained by scattering neutrons in a crystalline solid; the peak is due to diffraction, and its position and width contain important information on the internal structure of the solid. However, the actual interpretation is immaterial for the present purpose. The essential point is that for a certain chosen setting of one independent variable (the scattering angle), we have measured the corresponding values of a dependent variable (the intensity), knowing the uncertainty of each measurement (the vertical lines through the points in the diagram).

In the simple problem of the length measurements that we discussed above, there was no question that the parameter we were seeking was the average value. In the present, more complex situation there is nothing as simple as an average value, and it is not at all obvious which information we can actually extract from our data. Generally, the data reduction relies on our "scientific" judgement in the sense that we need to assume that our data "describe" a certain theoretical or analytical curve $y = F(x)$. In some cases we believe that our data should fall on a straight line, or a parabola, or the like. We then try to adjust a straight line or a parabola to see if we can get it to fit the data. This is exactly what a least-squares fitting program does. A straight line has two parameters, e.g. a slope and a cutoff, and the program varies these parameters until it finds the values of the parameters giving the least sum of the squared deviations. It might still give a poor fit, but we have criteria (discussed later) that will tell us whether the fit is statistically acceptable or not. If it is unacceptable, we have to think of a better analytical curve and try that instead.

In our example of a diffraction peak, we have reasons to believe that the data fall on a so-called Gaussian curve:

$$y = F(x) = I_o \exp\left(-\frac{(x - x_o)^2}{\eta^2}\right) . \qquad (1)$$

This equation contains three parameters: the peak or maximum intensity I_o, the peak position x_o, and the peak width η (the full width at half maximum is $\eta\, 2\sqrt{\ln 2}$). A least-squares program will vary the three parameters until it finds the "best fit" characterized by the least weighted mean square deviation,

$$X^2 = \frac{1}{N - 3} \sum_{i - 1}^{N} w_i (y_i - F(x_i))^2 . \qquad (2)$$

"N − 3" is the number of data points minus the number of parameters in the fit. We refer readers to standard statistical textbooks for a discussion of why this factor appears rather than N itself. The curve in the diagram is the best Gaussian fit to the data. Later on we shall return to this example and show how we judge the goodness of the fit and how we present the resulting parameters.

We would like to repeat, however, that we could have chosen other functions F(x) with fewer or more parameters and perhaps have found equally good fits. The reason for our choice is that we have a theory which predicts that F(x) should be a Gaussian function, and we have proved that the experimental results are in statistical accordance with this prediction. This shows clearly how data reduction is an interplay between theory and experiment.

TH[illegible]

Let us assume that we have measured N data points [X(I), Y(I)], I = 1, 2..., N, where X is the independent and Y the dependent variable. We consider X(I) to be known precisely* and Y(I) to be uncertain by $\pm\sigma_I$. Then the weight of the I'th data point is

$$W(I)\ 6\ \sigma_I^{-2} .$$

The fitting function is denoted $F_{\vec{p}}(X)$, where the "vector" $\vec{p}$ stands for the set of parameters

$$\vec{p} = [p(1), p(2), \ldots, p(M)]$$

which is to be adjusted to give the best fit of F to the data points.

As mentioned above, the best fit is characterized by an absolute minimum of the weighted mean square deviation,

$$X^2 = \frac{1}{N - M} \sum_{I=1}^{N} W(I) [Y(I) - F_{\vec{p}}(X(I))]^2 . \qquad (3)$$

Considering X^2 a function of the parameters $\vec{p}$, a condition for minimum is that

$$\frac{\partial (X^2)}{\partial p(K)} = 0 \qquad \text{for } K = 1, 2, \ldots\ldots, M. \qquad (4)$$

For polynomial regression or Fourier analysis, where the p(K)'s are simply coefficients in a series expansion or a trigonometrical series, and also for all other fitting functions which are linear in the p(K)'s, the optimal parameters can be expressed in closed form. That is, the M equations (4) become a set of linear equations which can be solved for $\vec{p}$ by standard techniques. In general, however, when F is not linear in all p(K)'s one must use iterational methods (note, for example, that F in Eq. (1) is only linear in I_o and not in x_o and η).

In the general case of a nonlinear F function, the idea is to approximate F by another function F′, which is linear. The linearization is perhaps best understood in pictorial phrasing. For a fixed X the function F describes a surface in the M-dimensional space of the parameters $\vec{p}$. When F is linear in the p(K)'s the surface is a plane; otherwise it is a curved surface. In the vicinity of any set of parameters, say $\vec{p}_o$, the tangential plane is the best "linear" approximation to the F-surface. Thus we shall replace the true F near $\vec{p}_o$ by its tangential plane, which has the analytical form

$$F'_{\vec{p} = \vec{p}_o + \Delta\vec{p}}(X) = F_{\vec{p}_o}(X) + \sum_{K=1}^{M} \left[\frac{\partial F(X)}{\partial p(K)}\right]_{\vec{p} = \vec{p}_o} \Delta p(K), \qquad (5)$$

where the derivatives

$$\frac{\partial F(X)}{\partial p(K)}$$

are evaluated at the values $p_o(K)$.

Of course, the best values of the p(K)'s are not known beforehand, but it is often possible to make a reasonable guess, at least to the right order of magnitude. Making a guess of all the parameters is the first step of the iterative procedure. Let the guess be $\vec{p}_o$. Using the linear F′ evaluated at $\vec{p}_o$, instead of F in Eq. (4), gives a set of linear equations which can be solved for all $\Delta p(K)$. These "corrections" are then added (or subtracted depending on their signs) to the guess parameters $\vec{p}_o$ to yield a "better" guess. This procedure is continued iteratively until two successive sets of guess parameters equalize to any chosen accuracy. With an accuracy of 1 per thousand and 3 parameters this happens typically after 3 or 4 iterations, depending on the closeness of the first guesses. It is worth noting that even wild guesses will in general converge towards the correct results, although exceptional cases could be envisaged where either a minimum is not found or it is not the absolute minimum.

We shall use the example from the introduction by fitting a Gaussian function to a set of data points X(N), Y(N), W(N) where X(N) is the independent variable, Y(N) the dependent variable and W(N) the uncertainty (standard deviation) in Y(N). Notice that W(N) is altered during calculation to the weight of the N'th point.

N	X(N)	Y(N)	W(N)
1	20.1	17	9
2	20.2	28	9
3	20.3	63	11
4	20.4	119	13
5	20.5	230	17
6	20.6	269	18
7	20.7	253	18
8	20.8	156	15
9	20.9	83	12
10	21.0	13	9
11	21.1	8	8

To illustrate the detailed processing of a run, we show in the following a listing using the PRINT ON option of the HP 9830, whereby all action is printed on the line printer. The orders and numbers keyed in and/or executed by the user are framed with full lines. The messages or questions from the calculator usually appearing only on the display are framed in dashed lines. The normal printout is left unframed.

*If X(I) is also uncertain by $\pm\sigma_I^X$, it is possible to transform σ_I^X graphically into an uncertainty in Y(I) which is added to the direct uncertainty in Y(I) (by root-squared summation).

To judge the statistical significance of the fit, the user must check the following items:

A. The reduced deviation must show no systematic trends, such as being positive in one end and negative in the other end of the abscissa range.

B. X^2 must be of order unity, or what amounts to the same; about 1/3 of the data points must have reduced deviations larger than unity and 2/3 of the data points reduced deviations smaller than unity.

In the example these conditions are fulfilled and we can state the results as follows:

The peak intensity $I_0 = 274 \pm 11$
(rounded off to integers)

The peak position $x_0 = 20.61 \pm 0.01$
(rounded off to 2 decimals)

The width parameter $\eta = 0.252 \pm 0.009$
(rounded off to 3 decimals)

The figure in the introduction shows the actual data points and the best fitting Gaussian curve.

Editor's note: A listing of "A Generalized Linear Least-Squares Fitting Program for the 9830A" by Dr. Dietrich and Mr. Rathmann may be obtained on request by writing to *KEYBOARD*, Hewlett-Packard, P.O. Box 301, Loveland, Colorado 80537, U.S.A.

OVE W. DIETRICH is a senior physicist with the Danish Atomic Energy Commission at the research establishment Risoe near Roskilde. He received his M.Sc. in physics from Copenhagen University in 1960 and became Dr.phil. in 1970. His major research interests are the applications of neutron scattering within solid state physics, in particular phase transitions and magnetism. From 1970 to 1971 he was a guest scientist at Brookhaven National Laboratory, N.Y.

OLE S. RATHMANN received his M.Sc. in physics from the Technical University in Copenhagen in 1973 and is presently working on his Ph.D. in the Physics Department at Risoe. His scientific interests lie within solid state physics, in particular magnetism and the study of condensed matter by neutron scattering technique.

```
LOAD0 EXECUTE
FETCH
10 DEF FNA(X)=P(1)*EXP(-((X-P(2))/P(3))↑2) END OF LINE
END
RUN EXECUTE
HEY,THERE,YOU GOT LSQ-FIT
DO YOU WANT USERS INSTRUCTIONS?
WRITE 1 FOR YES, 0 FOR NO?0
FIT EXPRESSION ENTERED?(1 OR 0)?1
RUN NO. = ?1
NO. OF PARAMETERS = ?3
GUESS VALUES OF PARAMETERS
P( 1    ) = ?, DELTA P( 1    )?270,.27
P( 2    ) = ?, DELTA P( 2    )?20.54,.02
P( 3    ) = ?, DELTA P( 3    )?.3,.0003
NOW DATA INPUT (LAST SET = 0,0,0)
 N                 X(N)             Y(N)              W(N)
X( 1    ),Y( 1    ),W( 1    )= ?20.1,17,9
 1                 20.1             17                  9
X( 2    ),Y( 2    ),W( 2    )= ?20.2,28,9
 2                 20.2             28                  9
X( 3    ),Y( 3    ),W( 3    )= ?20.3,63,11
 3                 20.3             63                 11
X( 4    ),Y( 4    ),W( 4    )= ?20.4,119,13
 4                 20.4             119                13
X( 5    ),Y( 5    ),W( 5    )= ?20.5,230,17
 5                 20.5             230                17
X( 6    ),Y( 6    ),W( 6    )= ?20.6,269,18
 6                 20.6             269                18
X( 7    ),Y( 7    ),W( 7    )= ?20.7,253,19
 7                 20.7             253                19
X( 8    ),Y( 8    ),W( 8    )= ?20.8,156,15
 8                 20.8             156                15
X( 9    ),Y( 9    ),W( 9    )= ?20.9,83,12
 9                 20.9             83                 12
X( 10   ),Y( 10   ),W( 10   )= ?21.0,13,9
 10                21               13                  9
X( 11   ),Y( 11   ),W( 11   )= ?21.1,8,8
 11                21.1             8                   8
X( 12   ),Y( 12   ),W( 12   )= ?0,0,0
BAD DATA SET NO.?7
X( 7    ),Y( 7    ),W( 7    )= ?20.7,253,18
 7                 20.7             253                18
BAD DATA SET NO.?0

LEAST SQUARES FIT  --  RUN NO.  1

ITER CHI↑2     PARAMETERS
                   1          2          3
  1 1.988E+01 2.700E+02 2.054E+01 3.000E-01
  2 2.068E+00 2.480E+02 2.061E+01 2.803E-01
  3 9.677E-01 2.705E+02 2.061E+01 2.527E-01
  4 9.497E-01 2.738E+02 2.061E+01 2.524E-01
  5 9.497E-01 2.738E+02 2.061E+01 2.524E-01

STD.DEV.        1.130E+01 6.854E-03 8.633E-03

DO YOU WANT A PLOT (0 OR 1)?0
 ABSCISSA     ORDINATE      CALC.       WEIGHT       RED.DEV.
 2.010E+01   1.700E+01   4.370E+00   1.235E-02    1.403E+00
 2.020E+01   2.800E+01   1.872E+01   1.235E-02    1.031E+00
 2.030E+01   6.300E+01   5.861E+01   8.264E-03    3.995E-01
 2.040E+01   1.190E+02   1.340E+02   5.917E-03   -1.153E+00
 2.050E+01   2.300E+02   2.238E+02   3.460E-03    3.657E-01
 2.060E+01   2.690E+02   2.730E+02   3.086E-03   -2.230E-01
 2.070E+01   2.530E+02   2.433E+02   3.086E-03    5.385E-01
 2.080E+01   1.560E+02   1.584E+02   4.444E-03   -1.594E-01
 2.090E+01   8.300E+01   7.532E+01   6.944E-03    6.399E-01
 2.100E+01   1.300E+01   2.616E+01   1.235E-02   -1.463E+00
 2.110E+01   8.000E+00   6.639E+00   1.563E-02    1.701E-01

NEXT PROBLEM
```

← Loading file 0 of the program.

← Keying in fit expression.

← Starting the program.

← Now input of data X(N), Y(N), W(N). You don't have to tell beforehand how many data points you have. When you are finished you just key in 0,0,0. Also, the input need not be in order of increasing X.

← The last number (19) in this line is erroneous; it should have been 18. But carry on and correct at the end.

← Now tell that No. 7 is wrong.

← And key in correct values.

← When all data are correct, reply Ø.

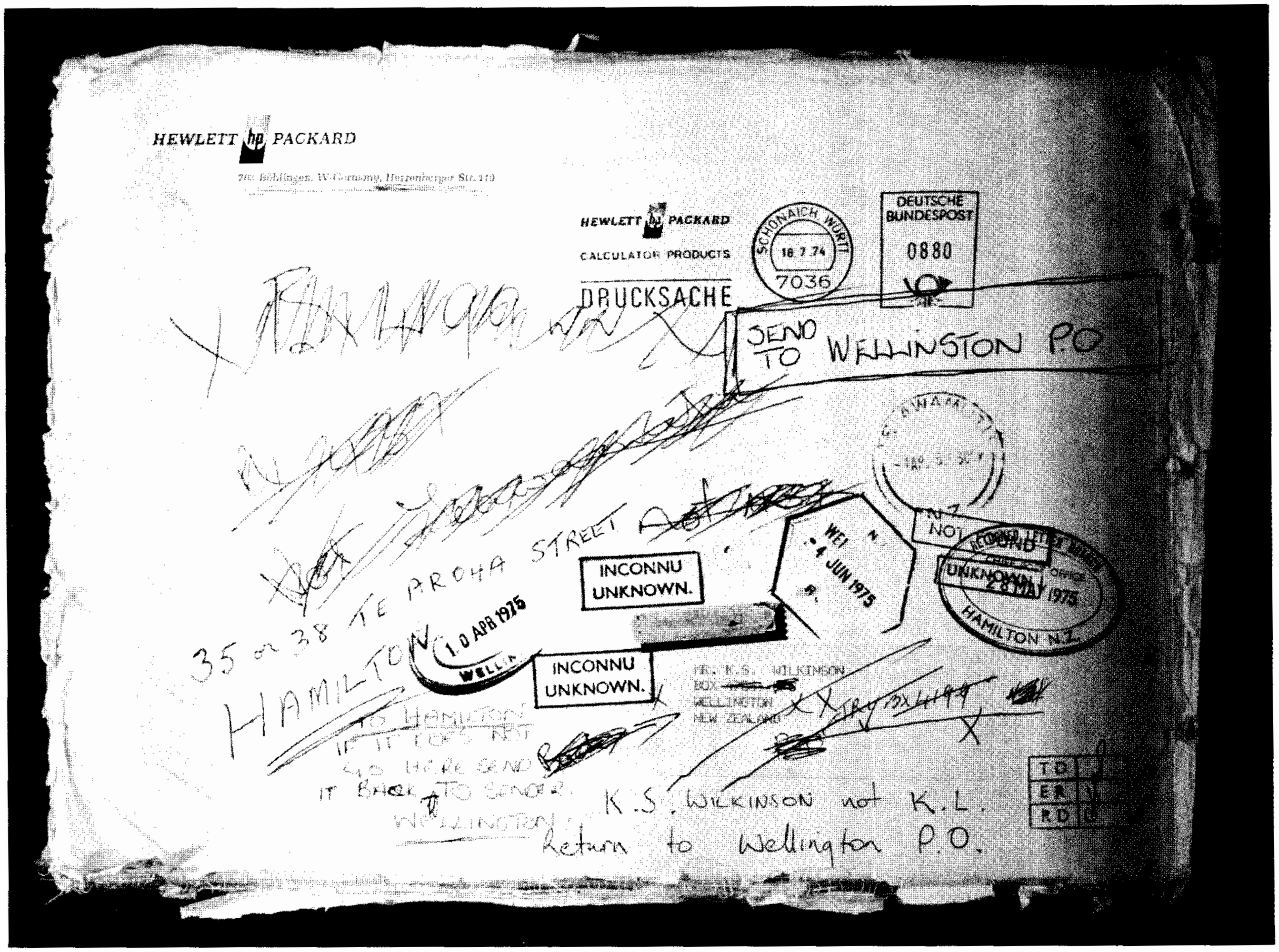
← Here computation starts, and file 1 is automatically loaded.

← The final values of χ^2 and the parameters.

← Standard deviation of parameters.

← Input and calculated data for all data points are printed.

← The program is ready for your next problem.



Last August this battered envelope was returned to sender, the 9820A/9821A Calculator Users Club in Boblingen, Germany, as undeliverable mail. The envelope was mailed in July, 1974. For over a year the New Zealand Post Office tried to deliver C.U.C. catalog updates to Mr. Wilkinson. As you can see from all the stamps and writing on the envelope, the Post Office must have exhausted all possibilities before returning it.

This is dedication to the job that we all can appreciate, especially those of us who make international mailings. Please check for clarity the manner in which your *KEYBOARD* is addressed. We sometimes have difficulty in correctly addressing non-U.S. mail because we are not familiar with all the international postal systems. But we try. And so do the postal services.

9830A [illegible]eds Pip[illegible] [illegible]me[illegible]

by Chuck Freeman
News Bureau Representative
Southern California Gas Company

Southern California Gas Company, largest gas distributing company in the nation, has more than 33,500 miles of distribution and transmission pipelines in the ground. In past years, the utility devoted thousands of manhours to scaling millions of feet of gas mains on tax rate and area maps by hand.

The constantly changing boundaries of tax and franchise districts as they expand, merge, or are newly formed, pose a particular problem for utility companies. Every foot of pipe representing a dollar investment in an affected area must be reported.

The traditional procedure involves placing a scale over the gas main on a tract map or atlas sheet, noting the footage and recording it, then totaling it at the end. The process was slow, tedious, and offered many opportunities for inaccuracies.

A "better mousetrap" appeared on the scene, however, when Southern California's engineering department procured a Hewlett-Packard 9830A Calculator with peripheral components. The system, originally purchased for engineering computations, consists of the 9830, a 9866A Thermal Printer for tabular output, a 9862A Plotter, and a 9864A Digitizer.

As various staff members became familiar with the operation of the equipment, they also became aware that the system was capable of a great deal more than was required of it. Practically any soluble problem presented and programmed properly was solved rapidly and accurately.

The engineering personnel assigned to tax and franchise matters started speculating on how the equipment could be programmed for scaling and tabulating underground piping within code areas. When their ideas were thoroughly jelled into a single plan, the plan was presented to Steve Schneider, an engineer with the utility who is adept at developing programs for the 9830.

Schneider developed a program that has proved highly successful. Several memory options have been added to the basic 9830 to increase its flexibility. These include a 4k memory, the String Variables, Extended I/O, and the Advanced Programming (APR I) ROM's. The String Variables ROM is required for handling alphanumerics for the purpose of displaying tax code numbers, the atlas sheet number, and footage.

The Extended I/O ROM is required to communicate with the Digitizer via the ENTER statement. To obtain greater flexibility in programming, the APR I ROM is used. Output from the program is in tabular form from the 9866 Printer.

Other capabilities added to the system include sub-tabulations, permanent storage (with atlas sheet numbers), remote keyboard typing, and plotting schematics of the piping networks that have been digitized.

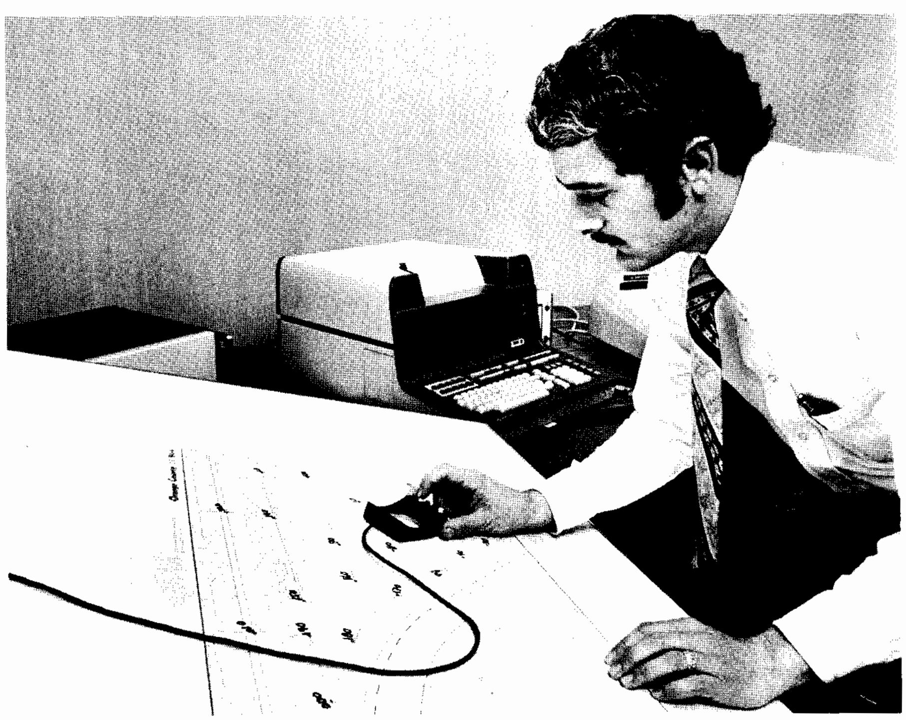
With the exception of map numbers and other identification input by the keyboard at the beginning of a work session, input to the calculator is mainly from the 9864 Digitizer. The Digitizer was ordered with an oversized 36-inch by 48-inch working area board — overall size is 42 inches by 54 inches. This size board makes the 9864 more suitable for use with large drawings and maps. The board is supplied ready to be bolted to a standard drawing table, but it was decided to mount it on a wheeled cart to allow some mobility.

An oversized cursor with a 2-inch diameter sight glass was also ordered to go with the board. Because of its size, it is easier to locate map points with the cross hairs, which is an important factor in expediting a project.

The Digitizer operates 100 percent under calculator control. When the 9830 Calculator requests data from the Digitizer, it receives an X-Y coordinate pair from the location of the hand-held cursor. Units of the coordinates, as they come from the 9866, are in inches. This simplifies the program, since it is easy to relate to a map via a simple feet-to-inches scale.

Lengths are calculated by the Pythagorean theorem. Since only relative length is calculated between two digitized points, the beginning and ending points of the pipe, document alignment on the Digitizer is of no importance.

The tax and franchise program is simple and straightforward (see flow diagram). A chief concern was that the program should be easy to learn for inexperienced operators.



OVERSIZED BOARD — Engineer Steve Schneider of Southern California Gas Company uses oversized cursor with two-inch glass to read points on tract map spread on oversized working area board tied to 9864 Digitizer. Digitizer feeds data into 9830 Calculator programmed for special tax and franchise purposes.

A "menu" approach to the Digitizer is utilized in the program. This is necessary since the calculator lacks any interrupt and cannot be redirected once a program is executed.

In the menu approach, a section of the board is reserved for special meaning. With the tax and franchise program, the menu calls for: pipe size (any number of sizes can be entered); continuous mode; error — delete last main measurement; summaries; and private franchise option.

The cursor works on four button commands: point of origin; single-point digitizing; automatic digitizing; and axis translation. There is a fifth button available but not used.

If a digitized point is detected within the menu, the program can determine the intended meaning and branch to take appropriate action. As an example, if a point is detected in the 1½ inch square, 2 inches x 6 inches right of the origin (0,0), then the operator commands a summary of the work completed on the map presently being worked on. The menu has the advantage that by modifying the geometric layout, the menu can be changed or expanded without major logic changes in the program.

As each pipe length is calculated, it is stored in an array according to size. The first version of the program had no provision for intermediate summaries, but this has since been added. This allows for tabulations of each map, as well as a grand total of pipe length by size at the end of a work session. Whenever a tabulation is desired, a point is digitized in the appropriate box on the menu.

Curved or meandering lines could create a problem, but, fortunately, the 9864 provides a continuous sample mode. To enter this mode, the operator hits a point in the continuous box on the menu. Then he presses the automatic digitizing button on the cursor and follows the curved line. The calculator now continuously requests and accepts data points. To exit, the operator presses the same button again and digitizes the appropriate pipe size on the menu. The length of the curved line is approximated by the many straight lines connecting the points on the line itself. The calculated length is stored as one segment.

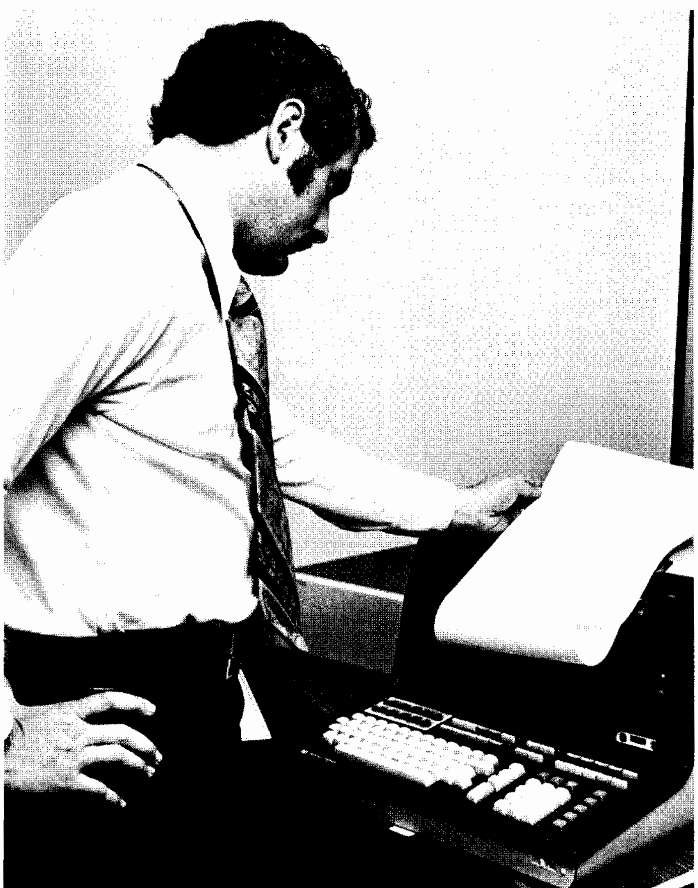
To avoid duplication of measurements, a clear overlay sheet is placed on top of the map and a felt pen is used to check off each main as it is measured.

The program has been honed to the satisfaction of all those involved for the moment, but improvements are constantly being suggested and incorporated into the system as new capabilities of equipment are realized. One such improvement is the addition of the capability to plot a schematic of all pipe measured on a map. This allows a quick visual check on which pipes were scaled.

As an example of how the program already has reduced manhours, the hand-scaling of 600,000 feet of main in the City of Irvine, California, required some 300 manhours. A similar project in Palm Desert, California, was performed recently using the new program to scale the same amount of footage in about 20 manhours.

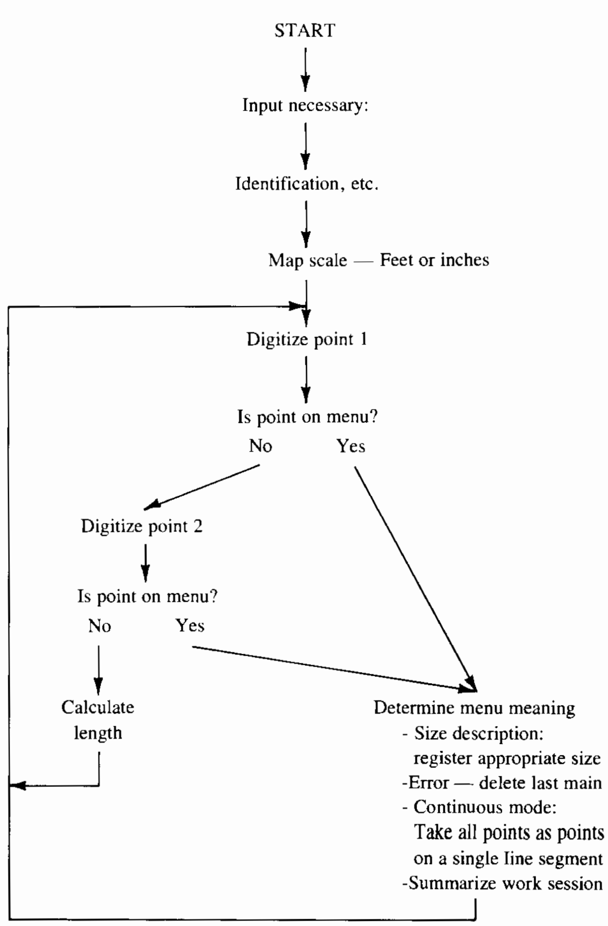
Under the old system, approximately five million feet of pipe could be scaled in a year. During the first month of the new program's operation, 1.5 million feet were scaled.

The capabilities of the Hewlett-Packard 9830 with its peripheral components are almost unlimited. All that is needed is a little imagination.



TABULAR OUTPUT — 9866 Thermal Printer is used with the 9830A Calculator by Southern California Gas Company for engineering computations.

BASIC PROGRAM LOGIC FLOW



PERMUTATIONS AND COMBINATIONS

By John Nairn, PhD
Hewlett-Packard Calculator Products Division

"If the older mathematics were mostly dominated by the needs of mensuration, modern mathematics are dominated by the conception of order and arrangement."

J. T. Merz

In the last few Crossroads articles we have been looking at the calculator as a tool for solving mathematical problems. One of the questions posed in that series concerned six men at a party, each taking a coat at random as he left. (Actually, they were quite inebriated, but there's no sense in bringing that up again.) The problem was to find the probability that at least one of the men would get his own coat. The solution involved enumerating all ways that the six coats could be distributed (a mathematician would say permuted) among the six men, then counting the number of ways in which one or more of the six got his own coat, and taking the ratio of the counts to be the probability.

Several readers knew that this approach would solve the problem, but did not know the method for calculating the ways of distributing the coats, and asked that the algorithm be given. As a result, this article will be devoted to the subject of permutations and combinations in general, and to algorithms for enumerating these arrangements in particular.

Before discussing anything, it is best if we know what we are talking about, so let's first define permutations and combinations. If I have three distinct objects (playing cards, for example) they may be arranged in six different orderings: namely, ABC, ACB, BAC, BCA, CAB, and CBA. These six arrangements are called the permutations of three things. Actually, the term "permutation" may be used more generally to mean the number of distinct arrangements of N objects chosen from a set of M objects (and can be made even more complicated when repetitions are allowed). But we will not get into that here.

Combinations are concerned with the number of ways that N objects can be chosen from a set of M objects, without regard to order. For example, the number of seven-person committees that could be formed from the U.S. Senate (consisting of 100 senators) would be a particular case of the number of combinations of 100 objects taken 7 at a time.

Although combinatorial analysis has become a powerful tool of modern science and mathematics, its origins can be traced back as far as the 8th century B.C. in China. The "I Ching" (or Book of Changes) is an ancient book of divination that uses 64 hexagrams in combination, with an elaborate ritual for choosing one, to tell the future. The hexagrams themselves represent the 64 permutations of three solid lines (the yin) and three broken lines (the yang). References to combinations are also found in the early Western world. The Greek philosopher, Xenocrates, calculated the number of possible syllables that could be formed from the Greek alphabet to be 1,002,000,000,000. (A result you may believe if you like!)

Many attempts were made through the centuries to obtain formulas that would give the number of permutations and combinations for a given number of objects. The goals of these calculations were quite practical and concerned such problems as the number of ways that the known planets could be taken two, three, etc., at a time for astrological purposes, or the number of combinations possible for a lock with several movable cylinders. It was not until the 17th century, however, that the work of such mathematicians as Fermat, Leibniz, and Pascal put combinatorial analysis on a firm foundation and established general formulas for their calculation.

The use of permutations and combinations can be broken down into two major classes: the first involves counting (or rather calculating) the number of permutations or combinations of N things taken M at a time; the second involves the actual enumeration of these arrangements. It is often sufficient to know only the number of possible arrangements, and indeed the solutions to many interesting mathematical problems are based on knowing how to find these counts. Many fields of modern science such as quantum physics and statistical mechanics are highly dependent on combinatorial analysis for their results.

The formulas for the number of these arrangements have been known for several centuries as:

$$P(n) = n! \quad \text{and} \quad C(n,m) = \frac{n!}{m!\,(n-m)!},$$

where P(n) is the number of permutations of n distinct objects, and C(n,m) is the number of ways of choosing m objects from a set of n distinct objects without regard to order. (Also, n! is read as "n factorial" and is the product of the first n integers.) Any introductory book on combinatorial analysis will give the derivations of these formulas and their extensions to cases involving allowed repetitions.

A more interesting task as far as calculators are concerned is the actual enumeration of these arrangements. In the problem mentioned at the beginning of this article, we can easily calculate (using the above formula) that the number of ways the six men could have taken the six coats is P(6) = 6! = 720. The answer to the problem is not obtained, however, until we actually enumerate the 720 arrangements and count the number of these for which someone got his own coat. For anyone with more ambition than I have (not a difficult condition to satisfy), this enumeration could be done by hand. But if the number of men involved were merely doubled, the number of cases to enumerate and test would jump to nearly half a billion! Obviously a job for a more patient and tireless enumerator — the trusty calculator.

Which brings us finally to the real purpose of this article, the presentation of two algorithms for the enumeration of permutations and combinations. At this point I am in the position of Buridan's proverbial ass which, when placed exactly between two equally delicious looking stacks of hay could not decide which to eat first, and so perished from starvation. Since my editor does not allow me the luxury of the do-nothing alternative, I must choose which algorithm to present first. And since the title of this article is permutations and combinations, I will begin with permutations.

In any computer simulation of a real problem, the first obstacle to be faced is that of finding a suitable notation. Returning to the problem of the six men and their coats, we would like to find all the ways that the coats can be distributed among the men. We could call the men Smith, Jones, etc., and the coats Smith's coat, Jones's coat, etc., but these would be difficult to work with in the program. A more realistic solution is to associate all of the men and coats with numbers (which the computer finds easier to digest). Let's associate the six coats with the numbers 1 through 6. If we now line up the owners of the coats in the same order, we may then look at all the permutations of the integers 1 through 6 and see how many of these arrangements match one or more of their owners. For example, Figure 1 shows two such arrangements of the six numbers representing the coats. Notice that the order of the men does not change.

Coats:	3 5 1 2 6 4	3 2 4 5 1 6
Men:	1 2 3 4 5 6	1 2 3 4 5 6

Figure 1

In the first case, no one got his own coat, whereas in the second arrangement two men got their own coats. Therefore our program needs only to enumerate all permutations of the numbers 1 to 6 and test whether a given number in the list matches its position in the list. For a general notation we will call the k^{th} element of the list a_k, where $k = (1,2,\ldots,n)$, and n is the number of elements in the list. The symbol $\{a_k\}$ will denote the entire list.

Never having found the time to develop my clairvoyant powers, I won't even try to guess in which language you would prefer to have me present the algorithms. So instead I will give them in the form of flow charts which can be easily translated into the particular language of your calculator. Figure 2 gives these flow charts for enumerating all of the arrangements of the first n integers, P(n), and all ways that m integers can be chosen from a set of the first n integers, C(n,m). I hope that these are self-explanatory with the exception of a few details which I shall expand upon now.

These are not flow charts for complete programs, but merely for those segments of a program to get the next permutation or combination for testing. In these flow charts the next arrangements are simply printed. In a working program, the box specifying printing the current list would be replaced with a more elaborate procedure in keeping with the purpose of the overall program. For example, in using the P(n) routine to solve the Coats problem, the print box would be replaced with a test of each element in the list to see if it matched its position in the list, and incrementing a counter if such a match is found. The first box in each algorithm represents an initialization of the list; i.e. setting $a_1 = 1$, $a_2 = 2$, etc.

The permutation algorithm is based on the fact that the n! arrangements can be broken down into (n-1)! groups of n arrangements each. These n arrangements within each group are merely the n arrangements obtained by rotating the group one element each time. The indices i and j are counters used to keep track of these groupings and rotations, and T is a temporary register used in carrying out the rotation.

The combination algorithm is simply a combinatorial odometer which keeps incrementing the last element of the list until it reaches the maximum number, n. When this happens, the next element to the left is incremented, the following one set to the previous one's new value plus one, and the whole process repeated.

These algorithms may not be optimal in some sense, and I would be interested to hear from anyone who can suggest improvements on them. Also, if anyone is using (or because of these algorithms can now use) permutations or combinations to solve actual problems, I would like to hear about your applications.

I would like to thank all of the readers who sent me their solutions to the problems posed in The Crossroads, Vol. 6, No. 4. Pierre Connolly sent a solution to the Harold's Army problem. Gus Hoehn and N. A. Barker each used a different novel approach to the Sailors and Coconuts problem which we don't have space to cover here.

By far the largest number of letters came in response to the Crossing the Desert problem. Michael L. Burrows, Harold R. Cheesman, F. C. Hulatt, Karel Kieslich, Robert L. Neal, and Dick Rahl all observed that the solution given (1962 miles) is far from optimal. Indeed, since the trip was finished with 38.1 miles of gas still left in the tank, any leg of the trip could be reduced by this amount. And since the first leg is repeated seven times, we can move all caches 38.1 miles closer to the starting point and cut off $7\times38.1 = 266.67$ miles. Thus the crossing can be made with a total of only 1733.33 miles driven. As Robert Neal observed in his letter, it is not surprising that the mathematician is out of work.

Finally I received a letter from Peter Gubis, Jiri Slavik, and Karel Vavruska in which they sent a packet of programs solving most of the problems posed, some in several ways. In particular, they solved the Coats and Drunkards problem we have been discussing in three different ways. One was by the evaluation of the analytic formula (see Crossroads 7-2). A second method used the enumeration of permutations that we have been discussing. And the third method used random numbers to simulate each man taking a coat as he left the party, and the program watched for someone to get his own coat. Their result for six men and 1000 trials gave a probability of 0.615 as compared to the analytic result of 0.632, which shows that there are many ways to skin a cat or clothe a drunkard!

Martin Gardner, "The Combinatorial Basis of the I Ching", SCIENTIFIC AMERICAN, Vol. 230 (1), p.108 (January, 1974)

C. L. Liu, INTRODUCTION TO COMBINATORIAL MATHEMATICS (New York: McGraw-Hill Book Company, 1968)

D. E. Smith, HISTORY OF MATHEMATICS, Vol. II (New York: Dover Publications, Inc., 1953)

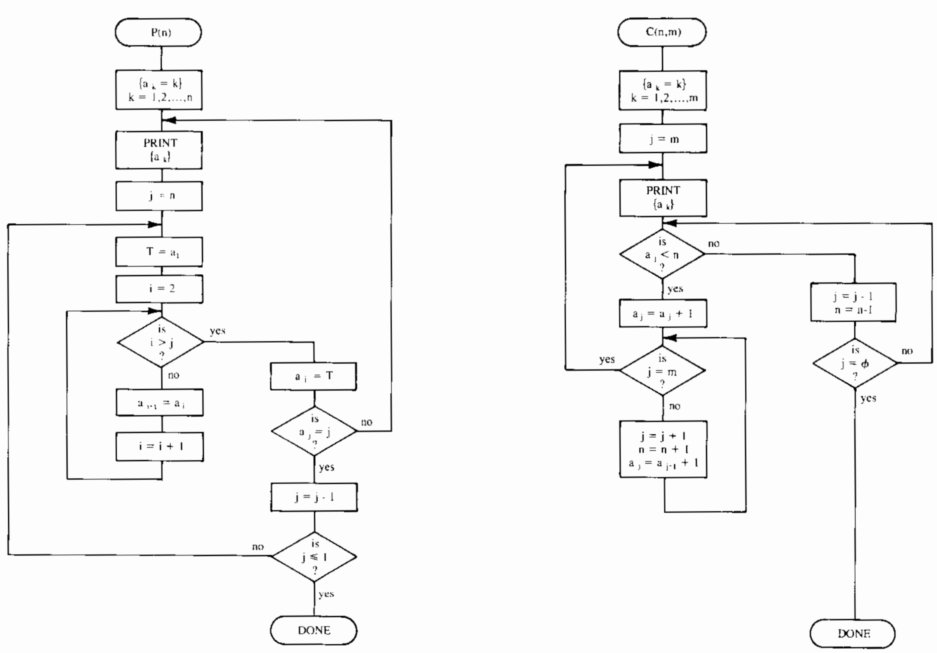


Figure 2: Algorithms for P(n) and C(n,m)

programming tips

Dennis Eagle of Calculator Lab, Hewlett-Packard, submits this programming tip.

There are times when it is desirable to be able to output characters which are not on the keyboard. For example, you might like to print 80 underscores across a page. If you had the underscore character, the problem could be solved by using a format statement in the following manner:

10 WRITE (15,20)
20 FORMAT 80"__"

Unfortunately, there is no underscore on the 9830A's keyboard. There is also no \, [,], line feed, or typewriter operations such as tab, backspace, etc. on the keyboard.

If you have a 9880 Mass Memory System, you can obtain the underscore and other non-keyboard characters as follows. First, execute the following instructions.

1. PRESS: FETCH
2. PRESS: f_0
3. TYPE: 1 DEF FNA (X)
4. PRESS: END OF LINE
5. TYPE: 2 STOP
6. PRESS: END OF LINE
7. END

Execute instructions 1 through 7 again, except that in instruction 2, press f_1. Execute instructions 1 through 7 for f_2, f_3, and so on up to f_9. Be sure that the two lines of programming are stored in every key. Next, key in the followng program:

```
10 DIM A$[80],Q$[1]
20 X=34
30 DBYTE X,Q$
40 A$="1GETKEY CHARKY "
50 A$[8,8]=Q$
60 A$[15,15]=Q$
70 FILES XXX
80 PRINT #1;A$,"2END","1RUN","1X=FNA1"
90 FOR N=0 TO 9
100 DISP "ASCII CODE";
110 INPUT X
120 IF X<0 OR X>127 THEN 180
130 DBYTE X,A$
140 A$[2]=A$
150 A$[1,1]="*"
160 PRINT #1;"1RUN","1DEL",A$
170 NEXT N
180 FOR N=N TO 9
190 PRINT #1;"1RUN","1","*"
200 NEXT N
210 PRINT #1;END
220 DGET"XXX"0
```

TYPE: SAVE KEY "CHARKY"
PRESS: EXECUTE
TYPE: OPEN "XXX",1
PRESS: EXECUTE
TYPE: SAVE "CHAR"
PRESS: EXECUTE
PRESS: RUN
PRESS: EXECUTE

ASCII CODE? will be displayed.

ENTER: 10
PRESS: EXECUTE

ASCII CODE? will again be displayed.

ENTER: 95
PRESS: EXECUTE

ASCII CODE? will be displayed. This time, terminate the program by entering 999.

ENTER: 999
PRESS: EXECUTE

The Mass Memory will make a few "clicking" sounds.

10 and 95 are the ASCII codes for line feed and underscore.

If you press f_0 three times, ⊦ ⊦ ⊦ will be displayed. Although the character is displayed as a ⊦, it will print as an underscore. Now whenever you want an underscore, you can press f_0.

TYPE: PRINT "ABC
PRESS: f_0
TYPE: DEF"

PRINT "ABC⊦DEF" will be in the display.

PRESS: EXECUTE

ABC_DEF will be printed.

As another example,

TYPE: 1 FORMAT 5"*⊦","test", 5" ⊦ ⊦"
(using the f_0 key for ⊦)
PRESS: END OF LINE
TYPE: WRITE (15,1)
PRESS: EXECUTE

Your printout should look like this:

```
*_*_*_*_*_TEST __ __ __ __ __
```

You can now type line feeds whenever you like. If you press f_5, J will be displayed. However, if the character is within a print or write statement, it will cause a line feed for the thermal printer or an index on the typewriter.

TYPE: PRINT "ABCJDEF" (using the f_5 key for J)
PRESS: EXECUTE

ABC
DEF will be printed on the thermal printer.

ABC
    DEF will be printed on the typewriter.

Pages F-6 and F-7 of the 9830A Operating and Programming Manual give all the ASCII codes and their corresponding outputs. The keyboard characters which can be stored are those with the following ASCII codes: Ø through 1Ø, 12, 14 through 31, 91 through 96, and 123 through 127.

The CHAR program above permits you to enter up to ten of these characters into the keys. For less than ten characters, terminate by entering 999.

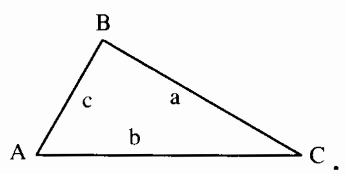
The characters are stored in the following sequence: f_5,f_0,f_1, f_2,f_3,f_4,f_6,f_7,f_8,f_9.

LAW OF COSINES (9100A/B, 9810A)

Our thanks to Robert L. Neal, Jr., Pacific Southwest Forest and Range Experiment Station, U.S. Forest Service, Challenge, California, for this programming tip.

The addition of "Change Sign" before "To Polar" in J. G. Langdon's Programming Tip for using the coordinate conversion keys of the 9100A/B and 9810A with Math ROM to solve the law of cosines (*KEYBOARD*, Vol. 6, No. 5, p. 14) will produce angle B in the y-register at the same time that side c is produced in the x-register. The remaining angle A of the triangle can then be solved by A = 180 − (C+B). Langdon's technique, modified as described, is used in the complete short program below, which solves for all unknown parts of a triangle given two sides and the included angle. With a, C, and b entered in x, y, and z at the first STOP, A, c, and B appear in x, y, and z at the second STOP. The area is produced in y at END. All angles are in decimal degrees.

```
00  1                   10  RUP
01  8                   11  CHS
02  0                   12  POL
03  XTO                 13  AC-
04   e  (b on 9810)     14  UP
05  STP                 15   e  (b on 9810)
06  AC-                 16  RDN (RUP, RUP on 9810)
07  RCT                 17  XEY
08  RUP                 18  STP
09  -                   19   f  (a on 9810)
0a  RUP                 1a   +
0b   X                  1b   2
0c  YTO                 1c  DIV
0d   f  (a on 9810)     1d  END
```



DISABLING THE END KEY (9830A)

Here's an interesting programming tip from Dr. John Roberts, Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California.

If you are a poor typist, as I am, you may find yourself pressing the END key instead of A, Q, or SHIFT with a considerably distruptive effect on whatever you are doing. The END key is one of the less used keys on the HP 9830 and its action can always be achieved by the combination of typing in END followed by EXECUTE. To disable the END key is a very simple matter. Bend a paper clip into a U and bend up the ends about 1/16 inch toward the middle to form a pair of hooks. Place this instrument around the key and gently pull it up. Put a simple 1/4 inch rubber grommet around the base of the key stem and then replace the key — preferably with END upside down to remind you (and the serviceman) that it is now inoperative.

TWO 9820A TIPS

Bob Huston of Surface Effect Ship Test Facility, Naval Air Station, Patuxent River, Maryland, submitted these two useful programming tips.

Following is some information we have discovered in our use of the HP 9820A Calculator with 9866A Printer, 9862A Plotter, 9869A Card Reader, and Peripheral Control ROM's I and II.

Use of Card Reader and Printer to List Cards

1. Load cards into reader.
2. Transfer 1,8 (PC II).
3. EXECUTE.
4. CONTINUOUS PICK (on 9869).

This will list cards on the printer. We have been using this feature to list 80-column cards containing Fortran programs. We have the punched card option on the reader.

Use of READ BYTE Key

If FMT "AD"; WRT 1 is executed, and then RDB 1 R(), a decimal code will be returned to the register that is the decimal equivalent of the ASCII. For instance, a space returns a 32, C is 40, 48 through 57 are digits 0 to 9, 65 through 90 are A to Z, and so forth. A 10 is found at the end of a card. By looping back to the RDB command and not the FMT, an entire card can be read in and decoded. This feature can be used to sort cards with the select hopper option on the card reader and will work on alpha or numeric data.

In our application we use the card reader and plotter to produce report-quality plots. In order to make the lettering of plots automatic, we use the routine mentioned above. All plot heading data and plot points are put on cards by a computer. The plots are done completely by the 9820, including lettering. Heading data is read into the calculator one column at a time. It is decoded using the short program given below and plotted using the plot commands of the PC I ROM.

We also use this method for special lettering of plots. It allows us to keypunch lettering and have the plotter produce high quality, finished work.

Printed in U.S.A.
December 15, 1975