

the I.P. Sharp newsletter

January/February 1981
Volume 9/Number 1



UNLOCKING THE I.P. SHARP PUBLIC LIBRARIES (The Glass Box Approach)

Lib Gibson, Toronto

I.P. Sharp provides a wide variety of applications software, ranging from data base management to statistics, from personnel systems to project management. This software is made available to our clients in the SHARP APL public library, organized in workspaces numbered under 500. In order to increase even further the usefulness of this software, we have decided to "unlock" the code in these workspaces. This means that any SHARP APL user will be able to display public library code. So much of the value of our application software at Sharp is dependent on our continuing enhancement and support that we feel we can take this unorthodox step. Whereas in the past a user could not display the code in these functions, now he has free access to see the code at any time.

This action will offer several benefits. It will be possible to tailor an existing system to satisfy the needs peculiar to an individual client. It also means that a module can be extracted from an I.P. Sharp system and incorporated into a customer's own system. So, if you like the user interface in one system and the data base management facilities of another, you will be able to conveniently extract

and incorporate them. The ability to examine the code will also be educational. Unlocked systems are like a glass box which can be studied for new techniques and approaches.

Of course, you must remember that these are in fact still proprietary functions of I.P. Sharp Associates, and are provided for the sole use of clients on the I.P. Sharp time-sharing system. Arrangements can be made to purchase or rent any of this software to run in-house. The price list is available from your local I.P. Sharp office.

Each of the functions we unlock will contain a line of the following form:

©COPYRIGHT I.P. SHARP ASSOCIATES LIMITED date©

It is illegal to remove this line from a function. We would also appreciate it if clients refrained from using a comment line beginning with ∇ as the first or last line of a function, since various utilities make use of this convention to determine whether a function is proprietary or not.

We will be unlocking our public libraries gradually over the next six months or so. We feel confident that the benefits to our users will be significant.

CONTENTS

Unlocking the Public Libraries	1
New Reference and Applications Library for Public Data Bases	2
New Toronto Stock Exchange Index and Stock Data	3
Air Transportation Seminar	4
Public Library Changes	4
COMPEC '80 Exhibition	5
People/Places	5

Technical Supplement 30

Greatest Common Divisor and Least Common Multiple	T1
A Letter to the Editor	T5
Dead Key Puzzle	T5
Contest 10 Results	T6
Practical Combinatorics, Part I	T6

NEW REFERENCE AND APPLICATIONS LIBRARY FOR PUBLIC DATA BASES

Ray Chantler, Toronto

Staying on top of the information contained in over 60 public data bases can be a problem. Users of the public data bases available on the I.P. Sharp computer system will be pleased to learn of the availability of a new workspace: 1 *DATABASES*. This workspace serves two purposes.

- It provides up-to-date reference material for each of the over 60 data bases which contain a total of more than 20 million time series.
- It contains a variety of pre-programmed reporting and plotting routines using data from many of these data bases.

The **reference library** is in fact a data base itself. That is, it is a data base of information on our data bases. Facts documented for each data base include a general description, history, frequency, sources, MAILBOX system contacts, on-line documentation, access, and references. Having such documentation on-line should enable both customers and staff to keep themselves informed about changes and additions to the public data bases as they happen.

The **applications library** is a collection of programs which use information taken from our public data bases. The programs are interactive and general, so that they can be tailored to individual needs. Those of you who are already using the aviation applications library will be familiar with this idea, which has now been extended across all categories of our data bases. The new library in 1 *DATABASES* includes programs which draw on data from the energy, financial, and economic data bases, along with the aviation programs. Some of the applications are:

- **Canadian housing starts and completions**, plotted by province (data from CANSIM).
- **U.S. imports of crude oil from OPEC and Iran** (data from PETROSERIES).
- **Daily close, market index, relative strength, and volume, in Canadian stocks**, for a chosen stock (data from the SECURITIES data base).

The programs in the data base applications library are the result of a cooperative effort between I.P. Sharp staff and customers. Many programs have been donated by our customers. If

you have a program you would like to have included in this library, please get in touch with us.

Following is a sample session using the Commodities data base.

SELECT 'EFUTURES'

MAIN PROGRAM IS: EFUTURES

FOR FULL DOCUMENTATION, TYPE: HOW
FOR A SAMPLE RUN, TYPE: SAMPLE

EFUTURES

COMMODITY ? (ENTER ? FOR A LIST; CARRIAGE RETURN TO EXIT)
COFFEE

ENTER NUMBER FOR FUTURE REQUIRED
(I.E. ENTER 3 FOR 3 MONTH FUTURES)

□:

6

DO YOU REQUIRE A TABLE, A PLOT, OR BOTH ? (T/P/B)

P

ENTER TERMINAL TYPE:

HP

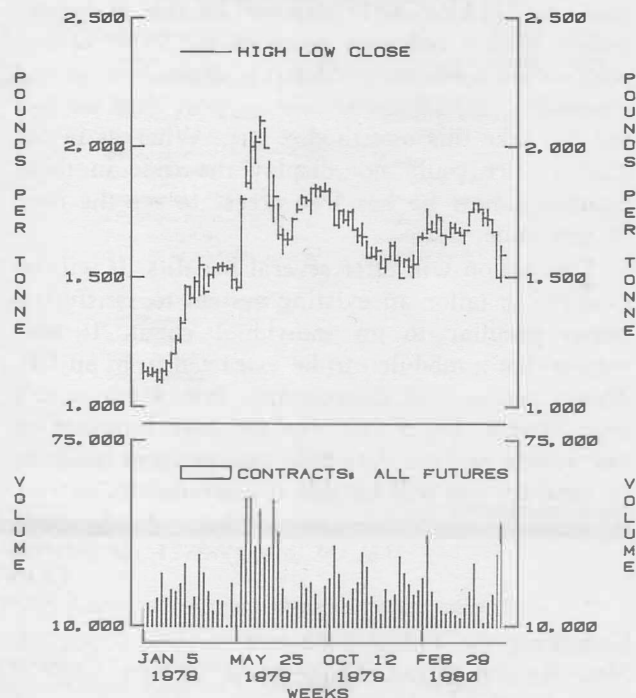
ENTER MONTHS OVER WHICH YOU REQUIRE THE WEEKLY SUMMARY
(E.G. 1 79 TO 4 79 FOR JAN+APRIL, 1979)

□:

1 79 TO 6 80

WEEKLY TRADING SUMMARY LONDON COMMODITIES EXCHANGE

COFFEE 6 MONTH FUTURES



***** PRICES QUOTED IN POUNDS STERLING *****

More information about the reference and applications library is on-line in the *DESCRIBE*.

NEW TORONTO STOCK EXCHANGE INDEX AND STOCK (TSE300) DATA

George Bihus, Toronto

Index data for the 62 Toronto Stock Exchange indices is now available for the period between 1956 and the present. Stock data, for the 300 stocks that make up the index system, is available from March of 1978 to the present. Twenty-four facts describe the TSE300 for the period from 1977, including a stock's high, low, close, volume, and value traded as well as base value, payout ratio, adjusted shares, and relative weight on the composite. From 1956 to 1976, the TSE300 is described by twelve facts, including yield, P/E ratio, aggregate earnings and dividends, relative strength, and quoted market value.

TSE300 data is available in three frequencies, daily from 1976, weekly from 1971, and monthly

from 1956. Data can be accessed for all frequencies with public workspace 39 *MAGIC* or 55 *RETRIEVE*. To obtain the on-line directory, load 55 *RETRIEVE* and type:

ΔVIEWDOC 'TSEINDEXHIST'

To access TSE300 data with 39 *MAGIC*, use *TSE300D* for daily data, *TSE300W* for weekly data, and *TSE300M* for monthly data. For further details on 39 *MAGIC*, consult the **MAGIC User's Manual**. The following example uses monthly data for the integrated oils index and its constituent stocks.

```
CLEAR
AUTOLABEL
AUTOTITLE
TITLE 'TSE INTEGRATED OILS INDEX AND CONSTITUENT STOCKS'
TITLE 'AS AT 31 OCTOBER 1980'
COLWIDTH 10
NOTIMESERIES
HIGHLIGHT
DECIMALS 3 3
MONTHLY,DATED AT 10 80
DISPLAY 'TXINO,BPO,GOC,HYO,IMO.A,MO,PFC,SHC,TPN,TXC' TSE300M 1 4 6 18
```

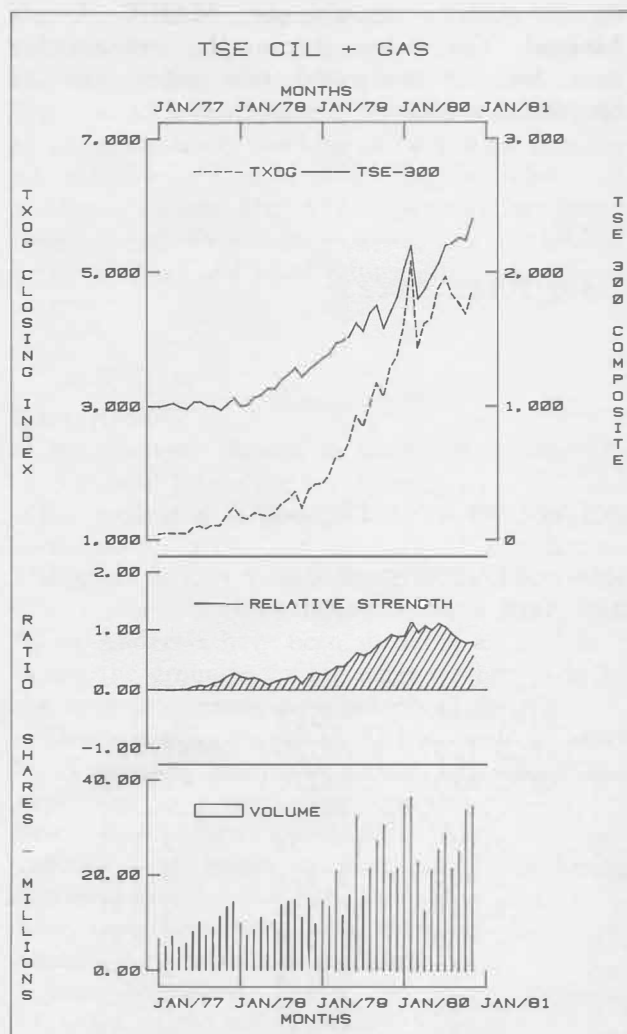
TSE INTEGRATED OILS INDEX AND CONSTITUENT STOCKS AS AT 31 OCTOBER 1980

	CLOSE	DIVIDEND YIELD (IN PERCENT)	PRICE EARNING RATIO	RELATIVE WEIGHT
TSE INTEGRATED OILS	2864.106	2.590	10.730	7.925
BP CANADA INC.	41.000	2.430	8.820	0.398
GULF CANADA LIMITED	24.250	1.810	15.580	2.831
HUSKY OIL LTD.	17.375	0.860	12.290	0.537
IMPERIAL OIL LIMITED CLASS A	37.000	3.910	8.570	2.306
MURPHY OIL CO. LTD.	36.500	0.000	23.380	0.066
PETROFINA CANADA INC.	75.000	2.500	11.330	0.308
SHELL CANADA LIMITED CLASS A	29.250	3.070	8.870	0.806
TOTAL PETROLEUM (NORTH AMERICA	27.750	1.720	9.320	0.286
TEXACO CANADA INC.	23.875	3.180	8.190	0.383

To access the TSE300 using *RETRIEVE*, define the view with the retrieve function *ΔVIEW*. For further details on *RETRIEVE*, consult the **RETRIEVE** manual. The following example shows a search for the symbols of the TSE Integrated Oils Index and its component stocks, as well as an access of their descriptions.

```
ΔVIEW 'TSEINDEXHIST'
ΔRESULTTYPE 'GLOBAL'
SYM←'OBJ' ΔSEARCH '///3 /01'
SYM ΔGET 11
```

The accompanying plot shows the large volume of trades of oil and gas stocks, and the effect of the price strength of these stocks on the oil and gas index and the TSE300 composite index.



AIR TRANSPORTATION SEMINAR

I.P. Sharp Associates will host an aviation seminar in New York on April 6 and 7, 1981, at the Halloran House in Manhattan. The theme of the conference is "Airlines in the 80's". There will be sessions on traffic, aircraft, financing and airline productivity, divided into four half-day sessions. The seminar has been organized for executives and professionals concerned with all phases of the air transportation industry.

We have had four successful aviation seminars over the past four years, and we expect the New York meeting to be equally successful. Attendance of approximately 100 professionals from the aviation community is anticipated.

We invite you and other members of your organization to attend and participate. There is no registration fee. Invitations will be mailed in January. For more information, please contact your local I.P. Sharp office.

PUBLIC LIBRARY CHANGES

Ed Stubbs, Toronto

New

- 1 *DATABASES* See Page 2.
- 1 *DIRECTDEF* This was 2 *DIRECTDEF*.

Changes

- 27 *HYBRID* A new function *MANUAL* provides a highspeed print of the user's manual.
- 32 *LAGS* 32 *PDLAG* has been renamed. It contains methods other than polynomial distributed lags.

Gone

- 81 *CSNEWS* It has been incorporated into 81 *CSUSAGE*.
- 1 *APL81* The deadline for submissions of abstracts for the STAPL-sponsored APL 81 conference has passed.

Technical Supplement-30

Greatest Common Divisor and Least Common Multiple now available as primitive functions

Paul Berry, Palo Alto

Two new primitives have recently been added to SHARP APL. They are LCM, which stands for "Least Common Multiple", and GCD, which stands for "Greatest Common Divisor". (In England, GCD is sometimes called HCF, for "Highest Common Factor".) GCD is denoted by the \vee symbol, and LCM by the \wedge symbol. Logical OR and logical AND (for which those symbols have long been used) are particular cases of the more general functions GCD and LCM. Gene McDonnell proposed the inclusion of GCD and LCM as APL primitives in a paper at APL-75 in Pisa.

GCD and LCM are scalar functions, so they can be applied to conformable arrays, and can participate in scans, reductions, and inner and outer products. For example, if X and Y are arrays of matching shape, then:

$X\vee Y$ GCD of X and Y
 $X\wedge Y$ LCM of X and Y

\vee/Y GCD of all elements along the last axis of Y
 \wedge/Y LCM of all elements along the last axis of Y

Concept of GCD: The GCD of a pair of numbers is the size of the largest unit in which they can both be represented as integers. GCD is especially useful in converting fractional data to a representation based on integers. In many applications, integers permit greater precision and more compact storage.

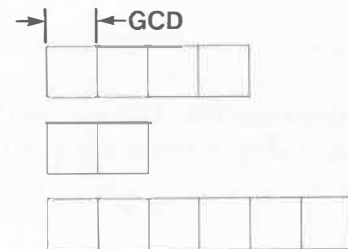
Suppose X is any number. A **divisor** of X (call it D) is a number which goes into X exactly N times. N must be an integer, but D is **not** required to be an integer. Then $N = X \div D$ and $X = N \times D$. Notice that N could be either negative or positive, depending on the sign assumed for D . To avoid ambiguity, GCD returns the **magnitude** of the largest divisor. Thus, $^{-}2\vee 8$ is 2 and $^{-}2\vee^{-}8$ is also 2.



A visual metaphor: Imagine the task of laying tiles in a long strip, perhaps the length of a narrow hallway. What length of tile will permit you to lay tiles end to end and exactly fill the hallway?

The length of each tile is the divisor D ; the number of tiles is N . Note that GCD does **not** tell you N (the number of tiles) but D (the length of each).

When you have only one number X (or only one hallway to tile) there's an infinite number of tile-sizes that will work. You could pick one enormous tile that just fits the entire space; or two, each of which covers half; or three, each of which covers a third; and so on indefinitely. When you have to use the same size tile in several **different** hallways, GCD is the longest tile you can use in **all** the hallways you are considering.



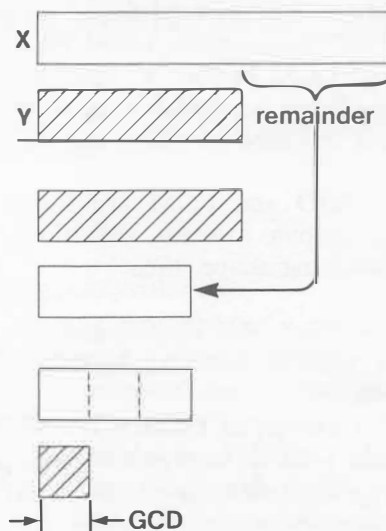
How the system evaluates GCD: The APL interpreter makes use of **Euclid's algorithm** to find the GCD of a pair of numbers. During reduction by GCD, it applies the same algorithm to the successive pairs. GCD is **commutative** (that is, indifferent to which value is the left argument and which is the right) and **associative** (that is, indifferent to the order in which a sequence of GCD's is evaluated).



Euclid noticed that a common divisor of X and Y must also be a common divisor of $X|Y$ (the remainder when X is divided by Y). Since the remainder $X|Y$ is smaller than either X or Y , repeatedly taking remainders converges on the result. Why the remainder can serve as well as the entire number is illustrated in the diagrams below and to the left.

Suppose you have two hallways to tile, one longer than the other. If the two hallways are side by side, the tiles in one must be in step with the tiles in the other. The end of the shorter hallway must fall opposite a tile-boundary in the other. But tiles of length D fit exactly in the total length of the longer hallway. So the remainder must also contain an integral number of tiles of length D . Euclid's algorithm for the GCD of two numbers α and ω can be summarized in the diagrams to the right, and in the following rules:

1. If α goes into ω an integral number of times, $|\alpha$ is the GCD.
2. Otherwise, replace ω by the remainder when ω is divided by α , reverse the roles of α and ω , and repeat.



As a program (in direct definition), that becomes $GCD: (\alpha|\omega) GCD \alpha : 0=\alpha|\omega : |\alpha$

Some Uses for GCD: Suppose you want to use integers to represent a set of quantities; the integers must be in the same proportion as the quantities. Divide each quantity by the GCD of all of them. Suppose the quantities are 6250, 8750, 11250, and 16250:

```
X ÷ v/X← 6250 8750 11250 16250
5 7 9 13
```

Or the quantities might be quite small and fractional:

```
X ÷ v/X← 0.03135 0.04389 0.05643 0.08151
5 7 9 13
```

Converting from "decimal" to fractional forms requires you to find two integers whose ratio is equal to the fraction. The two integers become the numerator and denominator in what is called a **rational approximation**. To measure the size of a fraction, compare it to 1, by finding the GCD of the fraction and 1. For a scalar fraction F , the expression is $X \div v/X \leftarrow F, 1$. Suppose the fraction is 0.265625:

```
X ÷ v/X← 0.265625 1
17 64
```

The function $RA: (\omega \circ . * 1 \ 0) \div F \circ . \vee 1 \ 1$ generalizes that procedure; it calculates a rational approximation for each element when ω is an array of fractions. Since the GCD is positive, the resulting denominators are always positive, and each numerator has the same sign as the corresponding element of ω . (The expression $\omega \circ . * 1 \ 0$ creates a new last axis containing each fraction and 1, since $\omega * 1$ is ω unchanged, and $\omega * 0$ is 1.)

Because of machine limitations on the precision with which certain fractions can be represented, many functions can calculate their results more precisely if they can be given integer arguments, even if only at intermediate stages of the calculation. Provided it is reasonable to think that a pair of fractional quantities have **some** common divisor, you may be able to increase precision by converting to integer representations. For example, if α and ω are large multiples of some fractional common divisor, the function $MINUS: ((RND \ \alpha \div GCD) - RND \ \omega \div GCD) \times GCD \leftarrow \alpha \vee \omega$ may compute their difference more precisely than $\alpha - \omega$ could. The expressions $\alpha \div GCD$ and $\omega \div GCD$ should both produce integers; the function $RND: L.5 + \omega$ simply makes sure that the system treats them as exact integers.

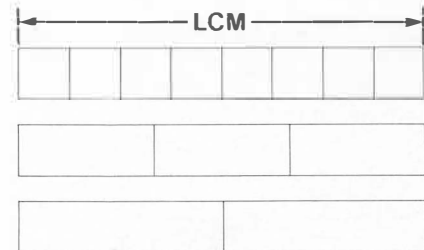
Similarly, matrix division is capable of greater precision when given integer arguments. You can convert the arguments to an integer representation, and then rescale the result by exploiting the identity $\alpha \div \omega \leftrightarrow ((\vee / , \alpha) \div \vee / , \omega) \times (RND \ \alpha \div \vee / , \alpha) \div RND \ \omega \div \vee / , \omega$.

Identity element for GCD: When you are finding the GCD of two numbers, if one of them is 0, the GCD is always the absolute value of the other. So for positive values, 0 is an identity element. When the domain is limited to 0 and 1, GCD gives the same result as logical OR.

Concept of LCM: Suppose you have a number X (which may be a fraction or an integer, and of either sign). A **multiple** of X is a number for which X is a divisor. Call the multiple M . Then M can be obtained by multiplying X by some integer N .

When you have a set X containing several numbers, a **common multiple** is a number for which **all** the members of X are divisors. There is an infinitude of such multiples; the **least** common multiple is the multiple having the smallest magnitude. The sign of the LCM is the sign of the product of the members of X .

Applying the visual metaphor: Whereas GCD finds the size of the longest tile that will fit exactly into hallways of various lengths, LCM finds the opposite. That is, you **start** with a pair of tile lengths, and find the shortest hallway that could be tiled using each of those lengths. (You have to use at least one tile, so hallways of length 0 don't count unless you have tiles of length 0.) If you lay down rows of tiles, one row for each length of tile, then the LCM is the distance you cover before all tiles are in step.



How the system evaluates LCM: The LCM of a set of numbers is their product divided by their GCD. Stated as a program (in direct definition) that is $LCM: \alpha \times \omega \div \alpha \vee \omega$.

Some uses for LCM: An application familiar from elementary arithmetic is finding a common denominator for a series of fractions. Indeed, the least common denominator **is** the LCM of the separate denominators. Suppose the following matrix F represents a set of fractions expressed as ratios. The top row contains the numerators, and the bottom row the denominators:

$$F$$

1	1	3	3	5	11	7
8	16	14	10	12	15	16

Then their common denominator is found by $\wedge/8\ 16\ 14\ 10\ 12\ 15\ 16$ (which is 1680). To convert the numerators, multiply each by the number of times its denominator goes into the common denominator; that is, by $(\wedge/F[2;]) \div F[2;]$. You could define a function *FCD* (for "fractions with common denominator") as *FCD*: $(CD \times \div f \omega), [\square IO-.5] CD \leftarrow \wedge/\omega[2;]$. It converts the array *F* into fractions having a common denominator, thus:

```

      FCD F
210 105 360 504 700 1232 735
1680 1680 1680 1680 1680 1680 1680

```

When two oscillations of different wavelengths are superimposed (for example, sounds, lights, radio signals, and so on) they are in step with each other at a length equal to the LCM of their separate wavelengths. This produces an interference pattern (or, in sound, a lower-pitched "beating" tone). If middle C and G above it have wavelengths of 10.8 and 7.2 cm respectively, sounding them together produces a beating wave whose length is $10.8 \wedge 7.2$ (which is 21.6 cm).

Sounds and lights are more usually identified by frequency than by wavelength, but since one is just the reciprocal of the other, the frequency of the beat between two frequencies α and ω could be found by *BEAT*: $\div (\div \alpha) \wedge (\div \omega)$. But the beat of two frequencies is their GCD; indeed, in general GCD and LCM for the domain excluding zero are related by a duality akin to De Morgan's law for logical OR and AND, thus:

$$\alpha \vee \omega \leftrightarrow \div (\div \alpha) \wedge (\div \omega)$$

Primes and factors: If a positive integer ω is prime, 1 is the greatest GCD it has with any of its possible divisors, so $1 = \omega \wedge \vee 1 \lfloor \omega * .5$. If the numbers in a vector ω are relatively prime (have no common factors), then it is true that $(\times/\omega) = \wedge/\omega$. However, $(\times/\omega) = \vee/\omega$ can be true only when ω is composed entirely of 1's, or has no more than one element.

Identity element for LCM: For any integer X , the LCM of 1 and X is X . So, for integers, 1 is the identity element for LCM. When the domain is restricted to the numbers 0 and 1, the result of LCM is the same as the result of logical AND.

Effect of comparison tolerance: As you might expect from the close relationship of GCD and LCM to residue, LCM and GCD are likewise subject to $\square CT$ (comparison tolerance). This has the practical benefit that GCD and LCM of various simple fractions are returned in a reasonable manner. For example, with $\square CT$ at its default value, $0.6 \wedge 1.4$ returns 4.2. It is impossible to represent such decimal fractions precisely within the computer. Given the problem $0.6 \wedge 1.4$, but working without tolerant comparison (or with $\square CT$ set to 0), the computer would fail to detect that both are divisible by 0.2, and hence would miss the result 4.2. Instead, it would return 3.02641895E16, which is no doubt a common multiple of the arguments, but by no means the least. The fact that "fuzzy" residue would help return more reasonable results for rationals that cannot be represented precisely in the computer's base-16 arithmetic was a contributing reason for making $\square CT$ apply to residue.

Acknowledgements: I am grateful to Doug Forkes, Simon Garland, Ken Iverson, Gene McDonnell and Ed Stubbs for many suggestions and corrections, and to Janet Okagaki for the diagrams.

Contest 11 Deadline Extended to March 1, 1981

Contest 11: Construct expressions which evaluate to give as many different natural numbers as possible, each containing precisely four 4's, but no other constants, and no other variables, and no user defined functions. Details in the November-December 1980 newsletter.

A LETTER TO THE EDITOR

Exponential Smoothing: "I found very interesting the essay by Robert Morris on exponential smoothing, in Technical Supplement 28. It is an excellent illustration of the power of APL, and the method should be useful in many forecasting applications.

As Dr. Morris is doubtless aware, exponential smoothing techniques of forecasting have their limitations as well as their virtues. The virtues are considerable. By use of a few simple lines of APL and with no exercise of judgement or trial and error one can get a forecast. Unfortunately, forecasts so generated are not necessarily very good: it depends on the nature of the series, and it is not a priori obvious which series will like this treatment and which won't.

My feeling is that if one is in doubt, one should begin by applying the Box-Jenkins technique. This is not a trivial exercise. It requires a great many lines of APL, at least in the version we have here, and quite a lot of time and judgement, but often it rewards one with excellent forecasts. One could then apply exponential smoothing to the same series and see if the forecasts were equally good, or almost as good. If they were, one could gain some confidence in the appropriations of exponential smoothing for the class of series under consideration, and thus have at hand a quick and easy forecasting method.

There is a discussion of this question in **Granger and Newbold, Forecasting Economic Time Series.**"

Louis Fulkerson
Computer Centre, Ahmadu Bello
University,
Zaria, Nigeria.

nameless, submitted an APL function giving the number of bytes used to store the numeric value, on the assumption that integers were somehow associated with four bytes, and non-integers with eight bytes. Of course this is not so. There are in fact many more representable integers on the SHARP APL system that require eight bytes of storage than there are that need only four: any integer larger than 2,147,483,647 will be stored as an eight-byte value, as will negative integers with magnitude larger than 2,147,483,648. Another contestant submitted a function which used event trapping to trap the domain error that would occur if the number, when used as the left argument to reshape, was a non-integer. Unfortunately the function also would cause a workspace full to occur for sufficiently large values of the argument, even though these were valid integers.

The answers which satisfied the statement of the problem are listed below.

```
DA:~x1|w ~x1Tω ~f1|w ~f1Tω ~xω-Lω
DB:0*1|w 0*ω-Lω
DC:~(fω)-Lω
DD:~fω-Lω
DE:0÷1|w
DF:0of1|w
DG:L1vω
DH:1+(f1|w)+1
DI:~0f-70,1|w
DJ:(fω)+1-Lω
DK:L200ω
DL:1/Δω,Lω (0-origin)
DM:1/Δ2+1|w (0-origin)
DN:1+Δ7ω,fω (0-origin)
DO:1+ΔΔ1,1200ω (0-origin)
DP:1/(Δ2+1|w)-1IO
DQ:0*1-L200ω
DR:0*1-TxT+200ω
DS:2|pp0,[1IO-1|w],10
```

Many of these functions depend on the basic property of an integer: it has a zero fractional part. I enjoyed *DE*, which exploits the definition APL gives to $0 \div 0$, and *DG*, which uses the newly added GCD-extension. The function *DJ* was not submitted as an answer, but was the source of the puzzle. It is found in section 3.3.3 of D.E. Knuth's **The Art of Computer Programming** (Seminumerical Algorithms). The funniest solution is *DS*, which exploits the difference between catenation and lamination.

Those who submitted answers were Larry Breed of IBM (*DI*), Roger Hui of the University of Toronto (*DA DB DC DE DF DH*), Oskar Itzinger of STSC (*DA DB DC DD*), Glen McDavid of Harris Bank, Chicago (*DD*), Bob Smith of STSC (*DL DM DP DS*) and Sandra Doubleday (*DA*), Jim Lucas (*DA DB*), Mickey Mehta (*DA DG*), Brian Oliver (*DA*), Roland Pesch (*DA DB DK DN DO DQ DR*), J. Henri Schueler (*DA*), and Gerard Westhoff (*DA*), all of I.P. Sharp, and Doug Holmes (*DA*).

DEAD KEY PUZZLE

Eugene McDonnell, Palo Alto

The dead key puzzle asked you for an expression which, given a numeric argument, had the value 1 if the number was an integer, and 0 otherwise, with the further constraint that all of the relational keys, and epsilon and iota, were unavailable. One contestant, who shall be

CONTEST 10 RESULTS

Congratulations to all who participated in contest 10 (see volume 8, number 4 for contest details). There were 55 entrants, from twelve countries, who submitted over 90 entries. The contestants were:

PETER AIRS, I.P. SHARP, LONDON
HENNING LERVAD ANDERSEN, NOVO INDUSTRI, COPENHAGEN
ROSS BETTINGER, HUGHES AIRCRAFT, EL SEGUNDO, CA
JOHN C. BOGAR, I.P. SHARP, WINNIPEG, SASK.
GARVIN H. BOYLE, STATISTICS CANADA, OTTAWA
ANDREAS BUCH, OR RESEARCH PROJECTS, ZURICH
TONY BUDD, BASILDON DEVELOPMENT CORP, BASILDON, ESSEX, UK
COLIN BURDEN, RANK XEROX, AYLESBURY, UK
JOHN CASSIDY, DALLAS COUNTY COMM. COLL. DIST., DALLAS, TX
RICHARD CHAMBERS, RANK XEROX, AYLESBURY, UK
PAUL CHAN, HERITAGE LIFE ASSURANCE, TORONTO
RAY CHANTLER, I.P. SHARP, TORONTO
ROBIN CHESTERS, I.P. SHARP, WINNIPEG, SASK.
LAWRENCE S. COHAN, POTOMAC, MD.
JERRY CUDECK, I.P. SHARP, TORONTO
M. DI FULVIO, B.C. TELEPHONE, BURNABY, B.C.
SANDRA DOUBLEDAY, I.P. SHARP, VICTORIA, B.C.
STEPHEN DURNFORD, LIVERPOOL, UK
DANIEL ESTES, DALLAS COUNTY COMM. COLL. DIST., DALLAS, TX
WOLFGANG E. FENDT, STIFTUNG REHABILITATION, HEIDELBERG
BILL FLOWERS, I.P. SHARP, TORONTO
LAURIE GELIATLY, I.P. SHARP, SYDNEY, AUSTRALIA
WAYNE HARRINGTON, DON MILLS, ONT.
GERALD HAUPTMANN, MOBIL OIL, VIENNA
RICHARD M. HEIBERGER, UNIV. OF PENN., PHILADELPHIA, PA
ROBIN HENNING, BURROUGHS WELLCOME, RESEARCH TRIANGLE PARK, NC
KIRK IVERSON, I.P. SHARP, TORONTO
S. KAHRKLING, LORETTA, ONT.
TOM KAROLY, CONFEDERATION LIFE INS. CO., TORONTO
RANDALL KAYE, ROYAL INSURANCE, TORONTO
PHILLIP KING, I.P. SHARP, TORONTO
JOSH LEVINE, I.P. SHARP, DALLAS, TX
FABIO MASSIMO, REGIONE LOMBARDIA, MILAN
DAVID MCKNIGHT, LE VESINET, FRANCE
ALESSANDRO MERAVIGLIA, REGIONE LOMBARDIA, MILAN
KEITH MORRIS, VICTORIA, B.C.
ALEX MORROW, PALO ALTO, CA
HAZEL O'HARE, I.P. SHARP, BRUSSELS
KEN PAWULSKI, I.P. SHARP, TORONTO
MICHAEL PETERSON, I.P. SHARP, DALLAS, TX
GARY POWELL, I.P. SHARP, SEATTLE, WA
NED ROBIE, TEXAS GULF, RALEIGH, NC
BRUCE ROE, DALLAS COUNTY COMM. COLL. DIST., DALLAS, TX
GREG SCHWARTZ, I.P. SHARP, PHILADELPHIA, PA
GLEN SEEDS, AMPEX, EL SEGUNDO, CA
JEFF SHALLIT, UNIV. OF CALIFORNIA, BERKELEY, CA
IAN SHANNON, ST. FAITH'S HOSPITAL, BRENTWOOD, ESSEX, UK
ROB SIDDALL, CONSUMERS GAS CO., SCARBOROUGH, ONT.
MARK SIMOS, PHILADELPHIA, PA.
BOB SMITH, STSC, BETHESDA, MD
DAVID GREER SMITH, I.P. SHARP, ROCHESTER, NY
GIB SMITH, IBM, ESSEX JUNCTION, VT
LOUIS VAN GELDROF, PHILIPS NETHERLANDS, EINDHOVEN
KEN WING, TREASURY BOARD, OTTAWA
YUEN K. WONG, TREASURY BOARD, OTTAWA

Many entries failed the first round of judging, which used a 5000-element vector, by giving a *WS FULL*. Those which survived were timed, with results as follows:

RANK	REL.TIME	ENTRY	CONTESTANT
1	1.000	VSUM	WAYNE HARRINGTON, DON MILLS, ONT.
2	1.059	CONTEW8	KEITH MORRIS, VICTORIA, B.C.
3	1.067	FUN	JERRY CUDECK, I.P. SHARP, TORONTO
4	1.091	CONTEST10	LOUIS VAN GELDROF, PHILIPS NETHERLANDS, EINDHOVEN
5	1.101	PRA7	PETER AIRS, I.P. SHARP, LONDON
6	1.363	MAC3	DAVID MCKNIGHT, LE VESINET, FRANCE
7	1.866	FLAM	ALEX MORROW, PALO ALTO, CA
8	1.887	FBOB	BOB SMITH, STSC, BETHESDA, MD
9	2.115	CNEDROBIE	NED ROBIE, TEXAS GULF, RALEIGH, NC
10	2.288	FN	ROB SIDDALL, CONSUMERS GAS CO., SCARBOROUGH, ONT.
11	3.983	MM	FABIO MASSIMO, REGIONE LOMBARDIA, MILAN
		AND	ALESSANDRO MERAVIGLIA, REGIONE LOMBARDIA, MILAN
12	13.067	RCH3	ROBIN CHESTERS, I.P. SHARP, WINNIPEG, SASK.
13	54.316	Δ2	GARY POWELL, I.P. SHARP, SEATTLE, WA
14	54.326	C	JOHN C. BOGAR, I.P. SHARP, WINNIPEG, SASK.
15	110.789	S3	GARVIN H. BOYLE, STATISTICS CANADA, OTTAWA, ONT.

The overall winner is **Wayne Harrington**, of Don Mills, Ontario. Congratulations! **Jerry Cudeck** of our Toronto office is the leading I.P. Sharp contestant. The winners will be notified soon regarding their prizes.

The two winning entries may be viewed in workspace 999 *CONTEST*. Thanks to all who participated.

PRACTICAL COMBINATORICS, Part I

Robert Metzger, Rochester

The title of this series is "Practical Combinatorics" for several reasons. Combinatorics is a branch of mathematics concerned with counting things. This series is "practical" because:

- 1) We suggest some everyday uses of combinatoric principles.
- 2) We present APL programs which solve some simple but useful combinatoric problems.
- 3) We study these programs in order to learn more about APL programming.
- 4) We use these programs as a means of introducing you to the Direct Definition notation for functions.

We begin by looking at an APL primitive function which you can use in combinatorial analysis.

The character **!** has several names. Some people call it "shriek", others call it "bang", and still others "quote dot". It's rather a neglected symbol and maybe you've never even used it. It represents both a monadic and dyadic function. Both functions are used in combinatoric applications.

The monadic function is called *FACTORIAL*. It is a scalar function. Its definition for scalars is given below.

FACTORIAL: \times/ω

This is a Direct Definition. The function name is separated by a colon from the statement which calculates the result. The ω stands for the value of the (right) argument. *FACTORIAL* multiplies together all the integers from 1 up to its argument.

One of the uses of the *FACTORIAL* function is in counting permutations. It can count all of the ways you can arrange all of the members of a set. If the set has N elements, you say $!N$. When you count permutations, a different ordering of the same objects counts as a different permutation. For example, $!3$ yields 6. The 6 permutations of the letters A, B, C are:

$ABC \quad BAC \quad CAB$
 $ACB \quad BCA \quad CBA$

FACTORIAL can be used to count permutations, but it does not actually create them.

When do you use permutations? An example would be the testing of a transaction driven system. Perhaps there are 6 types of transactions, which can happen in any order. Then there are $!6 = 720$ possible orders in which the transactions may be entered. If the effect of the transactions depends on the order in which they happen, you need to test all of the permutations. **If order is important, permutations may be what you need.**

We need to write an APL function to create all of the permutations of N items. The functions listed below give all of the possible permutations of ω , when $\square IO \leftarrow 0$. (Add 1 to the result for $\square IO \leftarrow 1$.)

```
PERMUTE1:  $\omega$  PERMROTATE PERMUTE1  $\omega-1$ 
          :  $\omega=1$ 
          : 1 1p0
```

```
PERMROTATE:  $(-1(\omega!\alpha)\div!\alpha-1)\Phi(\alpha-1),((!\alpha),\alpha-1)\rho\omega$ 
```

You have probably noticed that these Direct Definitions look different from the first one. *PERMUTE1* has a name, plus three APL statements. The first statement returns the result of the process while it is going on. The second statement tells you when the process will stop. The third statement returns the result of the process when it has completed. In *PERMROTATE*, α stands for the value of the left (first) argument. As with

FACTORIAL, ω stands for the value of the right (last) argument.

Let's look more closely at these functions. *PERMUTE1* is a recursive function. It invokes itself. Symbolic manipulation is a good use of recursion [see Technical Supplement 25]. This is a major part of combinatoric computing.

Some people feel that building recursive functions is hard. Direct Definition makes it much easier. This is because it provides a structure especially suited for recursion. One idea which makes recursion easier to work with is to separate the recursive process into at least 2 functions. The only purpose of the main function is to call itself until it decides that it has been called enough times. Then it returns a kernel of information which is used later. As each invocation of the main function completes, it gives what has been done so far to a second function. This second function actually does the work which builds up the result. This is how *PERMUTE1* and *PERMROTATE* work together. To generate all the permutations of 0 1 2, you say *PERMUTE1* 3.

There are some other important things to note about these functions. They reach *WS FULL* rather quickly. This is because $!N$ grows so quickly. They are quite efficient. This is because although the number of permutations grows as $!N$, the number of recursive calls grows only as N , that is, linearly. The algorithm can be considered an APL variant of the "Nested Cycles" algorithm, and is modified from a function in "Programming Style in APL", by K. Iverson, found in the proceedings of the 1978 APL User's Meeting.

This algorithm uses one APL feature that some people might not be aware of. Not only can you do scalar rotations like $3\Phi 15$, but also higher order rotations like $1\ 2\ 3\Phi 3\ 4\rho 12$ as well. For example, when the left argument to *PERMROTATE1* is 3, and the right argument is

```
2 0 1
0 2 1
1 2 0
1 0 2
0 1 2
2 1 0
```

and the original argument to *PERMUTE1* was 4, then the vector used in the rotation would be

```
0 0 0 0 0  $\overline{1}$   $\overline{1}$   $\overline{1}$   $\overline{1}$   $\overline{1}$   $\overline{1}$   $\overline{2}$   $\overline{2}$   $\overline{2}$ 
 $\overline{2}$   $\overline{2}$   $\overline{2}$   $\overline{3}$   $\overline{3}$   $\overline{3}$   $\overline{3}$   $\overline{3}$   $\overline{3}$   $\overline{4}$ 
```

This would rotate the matrix created from $3, 24 \ 3\rho$ of the right argument. The moral of this story is simple. If you know how to use an APL function with scalars and vectors, it is frequently valuable to learn how to use it with arrays of higher rank.

Let's look at another algorithm for computing permutations. Here the algorithm works with $\square IO \leftarrow 1$. To create all the permutations of 1 2 3 4, you say *PERMUTE2* 4.

```
PERMUTE2: PERMBUILD PERMUTE2  $\omega-1$ 
          :  $\omega=1$ 
          : 1 1  $\rho 1$ 

PERMBUILD:  $S\rho\omega$  PERMININSERT  $\omega$  PERMDIAGONAL  $S \leftarrow$  PERMSHAPE  $\omega$ 

PERMSHAPE:  $(\times/0 \ 1+\rho\omega), 1+\bar{1}+\rho\omega$ 

PERMDIAGONAL:  $\omega\rho(1+2\rho\bar{1}+\rho\alpha)\rho 0, (1+\bar{1}+\rho\alpha)\rho 1$ 

PERMININSERT:  $P \Delta P[(\sim\omega)/\bar{1}\rho\omega]+1+\bar{1}+\rho\alpha \Delta P \leftarrow \omega \setminus, (1+\bar{1}+\rho\alpha)/\alpha$ 

 $\Delta$ :  $\alpha$ 
```

Once again, the main function is recursive. *PERMBUILD* does the actual work. It relies on three subfunctions. All Direct Definitions return results, and normally take arguments. This makes it possible to compose them into "phrases" of subfunctions. This is the case with *PERMBUILD*. In Direct Definition, all variables which are assigned (like *S*) are automatically localized.

PERMDIAGONAL is a special case of the idiom for creating an $N \times N$ identity matrix: $(N, N)\rho 1, N\rho 0$. In this case we have 0's in the main diagonal, and 1's elsewhere. *PERMININSERT* actually has 3 statements, connected with the function Δ . This is necessary because only indexing operations are allowed to the left of the assignment arrow, and indexed assignments pass along only the changed elements. Hopefully, future versions of APL will correct this problem. Notice that replication is used in the rightmost statement to repeat rows.

This second algorithm relies on insertions, rather than rotations. An example of the transformation it performs at each level is shown below.

```
1 2 3 --> 1 2 3 --> 0 1 2 3 --> 4 1 2 3
          1 2 3      1 0 2 3      1 4 2 3
          1 2 3      1 2 0 3      1 2 4 3
          1 2 3      1 2 3 0      1 2 3 4
```

The insertions are performed by expanding the array, and then assigning into the expanded data. This is a rather standard way of doing insertions in APL. This algorithm can be considered an APL variation on the "Transposition of Adjacent Elements" algorithm.

If you run these functions, you will see that neither of them produce the permutations in "lexicographic order". The permutations of 1 2 3 in lexicographic order would be:

```
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

While there are algorithms to produce permutations in this increasing numerical order, in SHARP APL, it is far more efficient to use one of the algorithms already described, and then sort the result. This is possible in SHARP APL because Δ will take arguments of rank >1 . So, if you need your permutations in lexicographic order, you can do the following:

```
 $P \leftarrow$  PERMUTE1  $N \diamond P \leftarrow P[\Delta P;]$ 
```

In this article, we have described how to count permutations, how you might use them, and how to generate them. We have also introduced most of the features of Direct Definition. The series will continue by looking at such concepts as combinations and tuples. If you would like to try the functions we have described, you can find them in the group *COMBINATORICS* in workspace 999 *TECHSUPP*. Questions and comments are welcome, and may be addressed to the author.

I.P. SHARP EXHIBITS AT COMPEC '80

David Chivers, London

COMPEC began as an annual exhibition of minicomputers and peripherals. Over the years, it has grown to fill London's Olympia exhibition centre, and extended its scope to become Britain's largest computer exhibition, attended by everyone who has an interest in computers—from programmable calculators to mainframes. This year there were 28,000 attendees.

It therefore wasn't surprising that among the first-time exhibitors at COMPEC '80 in November, you could find both IBM, who took one of the smallest stands to show the 3101 ASCII terminal, and I.P. Sharp Associates, who occupied the largest stand of the whole exhibition. No, this wasn't a case of an excessive budget, but an opportunity to take part in a promotional campaign organised by the Canadian Department of Industry, Trade and Commerce on behalf of Canadian industry. The Canadian Government organised and paid for the stand, which I.P. Sharp shared with Volker-Craig, GEAC Computers, and two suppliers of micro-based APL systems, MCM Computers and Telecompute.

The I.P. Sharp team, ably lead by James Sinclair, demonstrated worldwide data base access, MABRA personnel systems and graphics with the HP7221A plotter.

Reprinted from **Computer Weekly**, 28.10.80:

"Certain trends have emerged on the software front; one particularly noticeable development is the rapid emergence and acceptance of APL under the guiding light of such organisations as I.P. Sharp. The growing band of APL enthusiasts will notice that it is far more in evidence at this year's COMPEC than ever before, and deservedly so. Its simplicity of use and speed of implementation render it an essential part of the scene."

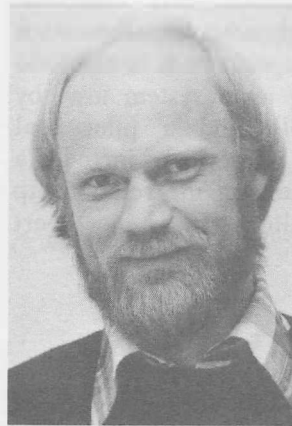
The following is an extract from an internal report of an organization who is a potential customer of SHARP APL on an in-house computer. This company is a long-time user of APL in a relatively remote part of the world.

We have, like a team of explorers, been bravely cutting our way through the undergrowth. Unbeknown to us, not far away, heavy traffic is pounding up and down on a freeway. Someone has done it all before. If we wish to make cheaper and more rapid progress to places we had not considered within our range, we must acquire a vehicle and get onto the freeway.

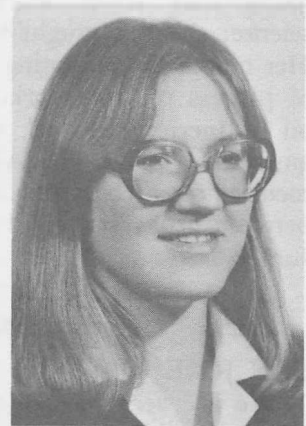
PEOPLE/PLACES

DUESSELDORF

Mike Zippel has joined our technical support staff in Germany. He studied physics in Kiel and at the University of Heidelberg, where he earned a degree in physics in 1973. He studied psychology in Mannheim and worked part-time as a teacher. In 1974 he joined the German Cancer Research Centre in Heidelberg, where he worked on the establishment of a spectroscopical data base, relating spectroscopical data, measured on 5 spectrometers (some of them on-line) to information about the chemical structure (molecular connection table) and physiological information. At that time, Mike learned the basics of APL.



Mike Zippel



Mari Vilmansen

MADRID: I.P. Sharp en España

I.P. Sharp opens an office in Spain.

Tenemos el gusto de anunciar la apertura de la oficina española de I.P. Sharp en Serrano 23, Madrid, cerca de la Plaza Colón. **Mari Vilmansen**, natural de Toronto, y que en los tres últimos años asistió al desarrollo de nuestra empresa en Estocolmo, se hará cargo de las operaciones.

Mari, licenciada en matemáticas y lingüística por la Universidad de Toronto, y que trabaja en nuestra empresa hace 7 años, ha aprovechado esta ocasión para añadir el castellano a los cinco idiomas que ya habla.

Hal Carim, ex-director de nuestra empresa en Suecia, le asiste en Madrid, encargándose del marketing del SHARP APL.

WHITE PLAINS

January 1, 1981 is the official opening date for the White Plains branch of I.P. Sharp Associates, Inc. The office in downtown White Plains will support New York's Westchester County, and Southern Connecticut.

Westchester County, dubbed "The Golden Apple", has become a major business and financial center. Its proximity to New York City has made it a good location for headquarters of over a dozen major corporations like Texaco, Inc. and General Foods. The Stamford, Connecticut area has experienced similar business and industrial growth and also hosts several corporate headquarters.

Kitty Orth of Carmel, New York will be working in the White Plains office in the areas of marketing and programming support. Kitty's background is in mathematics and computer science; she has taught high school mathematics. Her non-APL computing experience is in the field of process control, working on a system used by oil refineries, a steel mill, and a pilot coal gasification plant. Her APL experience includes a medical research system studying the relationship between drugs and renal failure after major heart surgery.

SAN JOSE

We are very pleased to announce the opening of our San Jose office on January 1, 1981. The office is located in Santa Clara, in the San Jose metropolitan area, and is part of the famous California Silicon Valley, home of the flourishing U.S. micro-electronics industry. **Diana Block** and **Chris Andrews** have been spending a lot of time laying the groundwork for what we hope will be the most progressive new branch of the 80's.

Diana is a former IBM APL'er with 13 years of experience there. She joined I.P. Sharp last September as a marketing representative for the new office. Chris joined I.P. Sharp late last summer and began training as a marketing representative for the San Jose office also. They have both been very busy learning about I.P. Sharp's data bases and packages.

June Sheppard, Pacific Northwest Regional Manager, is the acting Branch Manager.

STUTTGART/BOEBLINGEN

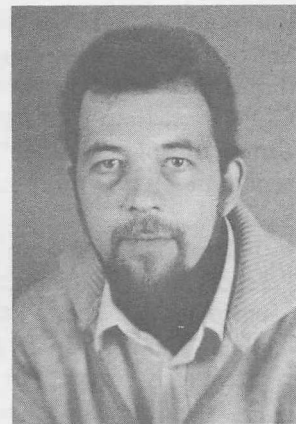
Conrad Hoesle has joined the company and will be Branch Manager of the Stuttgart/Boeblingen office. He was born in Stuttgart and graduated from high school in 1966. After leaving the University of Stuttgart, he changed over to Engineering College, (Fachhochschule fuer Technik in Mannheim), and finished with an Ing. (grad) fuer Chemische Technik (Engineer for Technical Chemistry) degree in 1971. Since then he has been working at the Nuclear Research Centre Karlsruhe, first in Radiochemistry, then in the analytical lab of the Water Technology Department. In 1975, Conrad took a course in APL and became a user of SHARP APL (which at that time was installed at the Research Centre). "Since I then was the only one with programming skills (however low) in the department, I got all the odd jobs connected with EDP."

In 1977, Conrad joined a newly formed department for the supervision of research and development projects, programming an APL system for financial and organisational control for the 250-odd projects for the department. "According to Parkinson's laws, what started as a comparatively simple task has since become a small department of its own."

During the last six months, Conrad has again been working in SHARP APL: "Of course, I was very impressed with the new possibilities such as `TRAP` and batch tasks." Conrad still lives in Stuttgart, about 20 kilometers from Boeblingen.

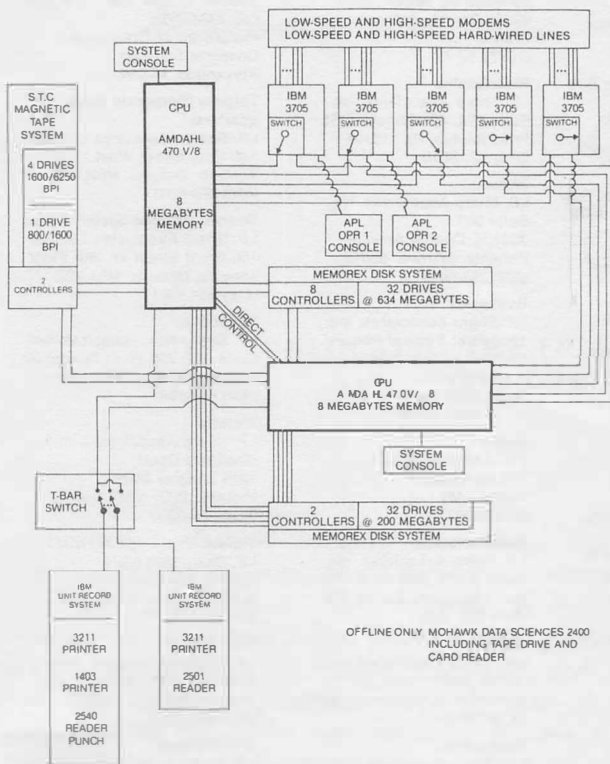


Kitty Orth



Conrad Hoesle

Hardware Configuration



SHARP APL COMPUTER CENTRE

Bill Apsit, Toronto

Hardware Changes

We recently replaced our remaining Amdahl V6-II computer with a second Amdahl V8. The new V8 has 8 megabytes of main memory, an increase of 2 megabytes over our previous system. We have also increased the main memory on the previous V8 to 8 megabytes as well. The diagram to the left shows the current hardware configuration of our Computer Centre in Toronto.

Hours of Operation Extended

SHARP APL service no longer shuts down on Saturdays. There is now only **one** scheduled shutdown per week, Sunday from 0400 to 1300 UTC. Customers on the west coast of North America will now have continuous service through Friday night, and European and Australian customers will have full Saturday service. Although service will now be available on Saturday, SHARP APL public data bases will not be available on that day from 0400 to 1200 UTC.

Comments? Questions? Suggestions?

The I.P. Sharp Newsletter is now into its 11th year of publication, with a circulation of over 5,000 copies worldwide. Over the years we have received many valuable comments and suggestions from our readers. We appreciate this feedback and encourage you to write to us. Please write to: The Newsletter, I.P. Sharp Associates, 145 King Street West, Toronto, Canada, M5H 1J8.

UPDATE

Label number: _____

- ☐ Please amend my mailing address as indicated.
- ☐ Please add the following name(s) to your Newsletter mailing list.
- ☐ Please send me a Publications Order Form.
- ☐ Please add my name to the Energy Newsletter mailing list.
- ☐ Please add my name to the Aviation Newsletter mailing list.
- ☐ Please send me information about your courses in

Name: _____

Title: _____

Co.: _____

Address: _____

(city) _____

Telephone: _____

The Newsletter is a regular publication of I.P. Sharp Associates. Contributions and comments are welcome and should be addressed to: Jeanne Gershater, I.P. Sharp Newsletter, 145 King Street West, Toronto, Canada M5H 1J8.

Jeanne Gershater, *Editor*
Ginger Kahn, *Assistant Editor*
Ginger Kahn, *Circulation*

Printed in Canada
January 1981
ISSN 0226 854X



I.P. Sharp Associates Head Office: 145 King Street West, Toronto, Canada M5H 1J8 (416) 364-5361

International Offices

Aberdeen

I.P. Sharp Associates Limited
5 Bon Accord Crescent
Aberdeen AB 12DH
Scotland
(0224) 25298

Amsterdam

Intersystems B.V.
Kabelweg 47
1014 BA Amsterdam
The Netherlands
(020) 86 80 11
Telex: 18795 ITS NL

Atlanta

I.P. Sharp Associates, Inc.
5000 Snapfinger Woods Dr.
Decatur, Georgia 30035
(404) 987-2301

Boston

I.P. Sharp Associates, Inc.
Suite 415
148 State Street
Boston, Massachusetts 02109
(617) 523-2506

Brussels

I.P. Sharp Europe S.A.
Avenue du General de
Gaulle, 39
1050 Bruxelles
Belgique
(02) 649 99 77

Calgary

I.P. Sharp Associates Limited
Suite 2660, Scotia Centre
700-2nd Street S.W.
Calgary, Alberta T2P 2W2
(403) 265-7730

Chicago

I.P. Sharp Associates, Inc.
2 North Riverside Plaza
Suite 1736
Chicago, Illinois 60606
(312) 648-1730

Cleveland

I.P. Sharp Associates, Inc.
(216) 431-6861
(local call, switched through
to Rochester office.)

Copenhagen

I.P. Sharp ApS
Ostergade 24B
1100 Copenhagen K
Denmark
(01) 11 24 34

Coventry

I.P. Sharp Associates Limited
7th Floor B Block
Coventry Point, Market Way
Coventry, England CV1 1EA
(0203) 21486/7

Dallas

I.P. Sharp Associates, Inc.
Suite 1148, Campbell Centre
8350 Northcentral Expressway
Dallas, Texas 75206
(214) 369-1131

Denver

I.P. Sharp Associates, Inc.
Suite 416
5680 South Syracuse Circle
Englewood, Colorado 80111
(303) 741-4404

Dublin

Gamma Data Systems Limited
(Agent)
Dollard House
Wellington Quay
Dublin 2, Ireland
(01) 711 877

Düsseldorf

I.P. Sharp GmbH
Leostasse 62A
4000 Dusseldorf 11
West Germany
(0211) 57 50 16

Edmonton

I.P. Sharp Associates Limited
Suite 505
10065 Jasper Avenue
Edmonton, Alberta T5J 3B1
(403) 428-6744

Gloucester

I.P. Sharp Associates Limited
29 Northgate Street
Gloucester, England GL1 2AN
(0452) 28106

Hamilton

I.P. Sharp Associates Limited
14 Hess South, Hess Village
Hamilton, Ontario L8P 3M9
(713) 526-5275

Houston

I.P. Sharp Associates, Inc.
Suite 375, One Corporate Square
2600 Southwest Freeway
Houston, Texas 77098
(713) 526-5275

London, Canada

I.P. Sharp Associates Limited
Suite 510, 220 Dundas Street
London, Ontario N6A 1H3
(519) 434-2426

London, England

I.P. Sharp Associates Limited
132 Buckingham Palace Road
London SW1W 9SA
England
(01) 730-0361
Telex: 8954178 SHARP G

Madrid

I.P. Sharp Assoc. Ltd.
Sucursal en Espana
Serrano, Piso 8
Madrid - 1, Spain

Manchester

I.P. Sharp Associates Limited
Paul House
89-91 Buttermarket Street
Warrington, Cheshire
England WA1 2NL
(0925) 50413/4

Melbourne

I.P. Sharp Associates Pty. Ltd.
520 Collins St., 13th Floor
Melbourne 3000
Victoria, Australia
(03) 614-1766

Mexico City

Teleinformatica de Mexico S.A.
(Agent)
Mail to:
Arenal N 40, Chimalistac
Mexico 20 D.F., Mexico
Shipments to:
A/C Despachos del Centro S.A.
Aeropuerto Internacional de la
Ciudad de Mexico
(905) 550-8033

Miami

I.P. Sharp Associates, Inc.
Suite D, Kennedy Building
14560 N.W. 60th Avenue
Miami Lakes, Florida 33014
(305) 556-0577

Milan

I.P. Sharp Srl
Via Eustachi 11
20129 Milan, Italy
(2) 221 612

Montreal

I.P. Sharp Associates Limited
Suite 1610
555 Dorchester Boulevard W.
Montreal, Quebec H2Z 1B1
(514) 866-4981

New York City

I.P. Sharp Associates, Inc.
Mail to:
Suite 2004
200 Park Avenue
New York, N.Y. 10166
Also at:
Suite 242, East. Mezz.
(212) 557-1200

Newport Beach

I.P. Sharp Associates, Inc.
Suite 1135
610 Newport Center Drive
Newport Beach, Ca. 92660
(714) 644-5112

Oslo

I.P. Sharp A/S
Dronningens gate 34
Mail to: P.Boks 486 Sentrum
OSLO 1, Norway
(02) 41 17 04

Ottawa

I.P. Sharp Associates Limited
Suite 600, 265 Carling Ave.
Ottawa, Ontario K1S 2E1
(613) 236-9942

Palo Alto

I.P. Sharp Associates, Inc.
Suite 201, 220 California Ave.
Palo Alto, Ca. 94306
(415) 327-1700

Paris

I.P. Sharp Sarl.
Tour Neptune, Cedex 20
20 Place de Seine
92086 Paris-la-defense
France
(1) 773 57 77

Philadelphia

I.P. Sharp Associates, Inc.
Suite 604, 437 Chestnut St.
Philadelphia, Pa. 19106
(215) 925-8010

Phoenix

I.P. Sharp Associates, Inc.
Suite 503
3033 N. Central Avenue
Phoenix, Arizona 85012
(602) 264-6819

Rochester

I.P. Sharp Associates, Inc.
1200 First Federal Plaza
Rochester, N.Y. 14614
(716) 546-7270
Telex: 0097 8473
0097 8474

Rome

I.S.I./Amdahl (agent)
Via del Corso 4
Rome, Italy
(6) 361-2997

San Francisco

I.P. Sharp Associates, Inc.
Suite C-415, 900 North Point St.
San Francisco, Ca. 94109
(415) 673-4930

San Jose

I.P. Sharp Associates, Inc.
3028A Scott Boulevard
Santa Clara, California 95050
(408) 727-9446

Saskatoon

I.P. Sharp Associates Limited
Suite 208, 135 21st Street E.
Saskatoon, Sask. S7K 0B4
(306) 664-4480

Seattle

I.P. Sharp Associates, Inc.
Suite 217
Executive Plaza East
12835 Bellevue-Redmond Rd.
Bellevue, Washington 98005
(206) 453-1661

Stockholm

I.P. Sharp AB
Kungsgatan 65
S111 22 Stockholm, Sweden
(08) 21 10 19

Stuttgart/Boeblingen

I.P. Sharp GmbH
Schafgasse 3
7030 Boeblingen
West Germany
(070 31) 2 30 14

Sydney

I.P. Sharp Associates Pty. Ltd.
Suite 1351, 175 Pitt Street
Sydney, N.S.W., Australia 2000
(02) 232-6366
Freight to: c/-Greenaways
Customs Services
Alexandria, N.S.W.

Toronto (Corporate Headquarters)

I.P. Sharp Associates Limited
145 King Street West
Toronto, Ontario M5H 1J8
(416) 364-5361

Toronto (Special Systems Div.)

I.P. Sharp Associates Limited
156 Front Street W., 5th Floor
Toronto, Ontario M5J 1G6
(416) 364-5361

Vancouver

I.P. Sharp Associates Limited
Suite 902, 700 West Pender St.
Vancouver, B.C. V6C 1G8
(604) 687-8991

Victoria

I.P. Sharp Associates Limited
Chancery Court
1218 Langley Street
Victoria, B.C. V8W 1W2
(604) 388-6365

Vienna

I.P. Sharp Ges.m.b.H
Rechte Wienzeile 5/3
A-1040 Wien, Austria
(0222) 57 65 71

Washington

I.P. Sharp Associates, Inc.
Suite 400, 1835 K Street N.W.
Washington, D.C. 20006
(202) 293-2915

White Plains

I.P. Sharp Associates, Inc.
180 East Post Road, LL-6
White Plains, New York 10601

Winnipeg

I.P. Sharp Associates Limited
Suite 208
213 Notre Dame Avenue
Winnipeg, Manitoba R3B 1N3
(204) 947-1241

Zurich

I.P. Sharp A.G.
Fortunagasse 15
8001 Zurich
Switzerland
(01) 211 84 24

SHARP APL Communications Network: Local Access Cities

APL OPERATOR VOICE (416) 363-2051 COMMUNICATIONS (416) 363-1832

Local dial access is available in all locations listed above. The SHARP APL Communications Network also provides local dial access in:

• Alliance • Ann Arbor • Birmingham • Buffalo • Clewiston (Fl) • Des Moines • Des Plaines • Detroit • Ft. Lauderdale
• Greenwich (Ct) • Halifax • Hartford • Hull • Kitchener • Laurel • Liverpool • Los Angeles • Lyndhurst • Minneapolis
• New Orleans • Oxford • Raleigh • Red Deer • Regina • Santa Ana • Sunnyvale • Syracuse • Towanda • Ukiah • Warrington
• White Plains (NY)

Our network connects with the Value Added Networks in:

• Alaska • Argentina • Bahrain • Bermuda • Finland • Hawaii • Hong Kong • Israel • Japan • Luxembourg • Mexico • New Zealand • The Philippines • Portugal • Puerto Rico • Singapore • Spain • Taiwan

In the continental United States the SHARP APL Network is interconnected with the Value Added networks to provide access in 170 more cities, and in Canada with 40 more. In all, with the 80 cities served by the I.P. Sharp Network listed above, SHARP APL is accessible from close to 300 places via a local phone call. Please ask at your nearest I.P. Sharp office for a complete list of access points and access procedures.