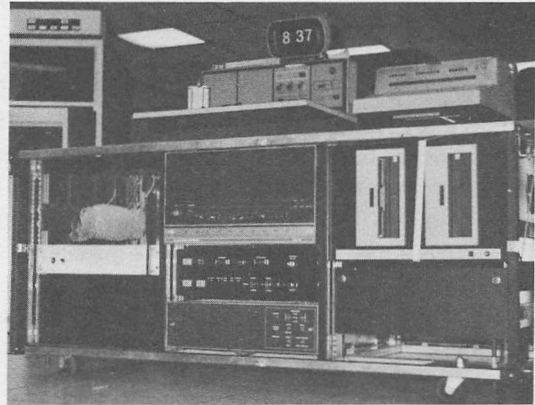


the I.P. Sharp *newsletter*

OCTOBER 1977
Volume 5 Number 5



RED PASSES SHARP APL TORCH ON TO AMDAHLV6

Eric Iverson
I.P. Sharp Associates, Toronto.

It was with some sadness on the 22nd of September that we powered off the RED IBM 360/75 for the last time. It has gracefully given way to the AMDAHL V6-II that arrived two weeks earlier. RED was a good, reliable machine that had done very well over the last two years for its many users. The V6, with a very smooth installation and a perfect week of production work seems a suitable successor.

The new machine will of course be sharing the load with the BLUE 75 that remains, although, at nearly six times the processing speed we expect the AMDAHL to bear the lion's share. This additional horsepower - along with other planned hardware and software changes - should substantially improve system performance as well as provide room for growth.

For the first few weeks the V6 will be connected to the production system as the SLAVE processor. A hardware fault on the SLAVE affects only those users on the SLAVE. When we have sufficient confidence that the V6 is at least as reliable as BLUE, we will give it a chance to be the MASTER. It's carrots like these that make machines behave!

Right now there is some disagreement in the APL development group as to whether in the long run the V6 would serve better as the MASTER or as the SLAVE. The suitability is a complex function of the user load and of fine tuning of the scheduler. Perhaps we will end up giving them turns!

COMPUTER SYSTEM PROVIDES INSTANT WORLDWIDE CONTACT

Thomas Williams, Sybron Corporation.

Sybron Corporation has implemented in APL an improved method of financial reporting, back and forth, between divisions and the corporate staff. According to Assistant Controller John M. Mitchell, Jim Torrens (Manager of Corporate Information Systems) is the person most directly responsible for establishing the Sybron Computerized Management Information system (MIS). Now Jim and his staff are responsible for maintaining, operating and expanding the system.

'MIS is just a trade word for a method of gathering information, storing it, and then being able to get at it when you need it,' Torrens explains. 'The file cabinets filled with paper, that preceded the computer system, constituted a management information system. It was much bulkier and it took a lot longer to find things, but it was an MIS,' he adds.

The system links 34 Sybron locations throughout the world. At each of the locations there is a computer terminal that serves as a transmitter and receiver. Those 34 terminals are connected over telephone lines to the Sybron corporate office. The corporate terminal is also linked to a computer (the SHARP APL system). Once a month, each division reports financial data - sales, orders, shipments, inventories, etc.

'Before the computer system was installed, the reports usually required 12 to 14 typewritten pages and were mailed to us. Mailing took four to five days, and that's airmail from overseas - if they didn't get lost in the mail or end up in a stack set aside in the Rome post office.' Once the reports were received, there still was work to be done. Computations had to be made, and certain items had to be correlated and pulled together for different types of reports.

That was then. Now, 'The four or five days mailing time has been reduced to seconds,' Torrens says. 'That gives people a little extra time to get their reports together.' The reports are then fed into the computer. 'The computer turns the data into information.' He estimates that the computations and organization of data done by the computer equals the work of 20 or 30 people over a period of a couple of days. From those 34 sets of figures, the computer prepares over 50 different types of reports used for financial analysis by the planning, budgeting, treasury, and controller's departments. 'In addition, the computer is set up to provide us with financial results reported by competitors. That means a division can compare its performance against that of other companies in the same market.'

Besides speeding up the reporting process, Torrens cites other advantages of the system: reduced paper, increased flexibility, confidentiality and greater ease and speed in retrieving information. 'Instead of digging through file cabinets for a report that might not be there, the person here or at one of the divisions who needs a certain piece of information can get it by punching the right keys on the computer terminal. It will be displayed on the cathode ray tube screen or typed out on paper depending on how the person wants it.'

At this point Sybron is looking forward to creating broader uses for the system that would incorporate types of communication other than strictly financial data.

Reprinted by permission from:

Sybron Reporter Volume 37 No. 4 - July/August 1977, with thanks.

SHARP PEOPLE AND PLACES

AMSTERDAM - From the first of October, **Laurie Howard** takes over as Manager of Intersystems. Laurie has been with the company for 6 years, so is no stranger to many of you.

The new address in The Netherlands is as follows:

Intersystems BV, - telephone (20)24 40 50
 Herengracht 244,
 1016 BT Amsterdam,
 The Netherlands.

LONDON -



Hazel O'Hare

Hazel O'Hare, previously Consulting Manager for the U.K., has become Branch Manager of our London office. She has a degree from the London School of Economics in Computing, and worked for several years around London University, doing survey analysis and computer user support. Hazel joined Sharp in 1975 and has worked on several large financial planning systems.

BRUSSELS - The correct address for the Belgian company is:

I.P. Sharp Europe S.A. - telephone 649 99 77
 Ave. General de Gaulle, 39
 1050 Bruxelles.

VIENNA - another correction: local dial access did not become available in September as was stated on the 'Sharp News' page last issue. The Network Map on page 7 was correct. Local access will however become available soon.

COPENHAGEN -



Margaret Reilly

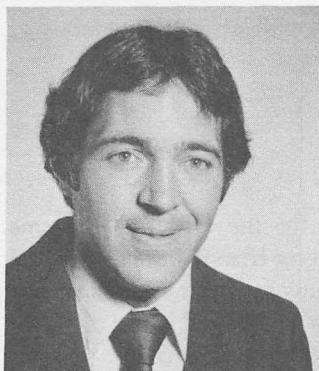
Johnny Larsen takes over as Manager of the Danish company from **Margaret Reilly**. Margaret will spend the next few months in Europe before returning to North America.



Johnny Larsen

SHARP PEOPLE (continued)

CALGARY - Clarke Bruce



Clarke joins **Jane Yates** in Calgary after three years with our Ottawa branch. His interest in financial applications (AIDS~AIDS) and his experience with government budgeting and accounting make him well suited to the Alberta region.

LETTER FROM LONDON

Valerie Lusmore,
I.P. Sharp Associates, London.



The lion and the unicorn, were fighting for the crown,
The lion beat the unicorn, all about the town.

(Traditional Nursery rhyme).

Well, our particular British Lion was the IPSA London office, which was involved in producing UNICORN, so that we could all go 'all about the town' together. What, you may well ask, is UNICORN?

It's the name of a planning system developed for the Wellcome Foundation Ltd (often known as Burroughs Wellcome) and is named after their corporate symbol. The company is a UK-centred, world-wide pharmaceutical company with sales in 1975/6 of around \$ 500 million. John White, a member of their Group Corporate Planning Directorate, recently gave a paper at IFIP 77 discussing planning and his experiences, and this set me to reviewing what we had accomplished to date.

The original task set us by Wellcome in early 1975 was to assist them to computerise their existing manual 3-year planning and budgeting system. This covered some 70 corporate entities and was a straightforward, comprehensive financial plan supported by marketing and manpower data. There were about 1000 data items, each for 10 time-periods. The system had the advantage of being known and understood throughout the organisation already, so the computerised version was readily acceptable, even to naive users of a computer system. It had the advantage of also being timely, the plan being presented to the Board of Directors three months earlier than in the previous year. This, despite unforeseen delays in the arrival of data by post from countries where one suspected the postal service ran by mules or pigeons!

The first year's run was done using the Sharp AIDS package and took only 2 months of effort from John White and myself, ably assisted by Tim Cureton, then a relative newcomer to IPSA. Other IPSA people wrote small sections and the accountants in the Wellcome planning department, many in their first venture with computers, wrote computer models using the AIDS commands. Much of the credit, however, for the success of the first system must go to Mr. Harry Copestick, a director of the Wellcome Foundation, who initiated the enterprise, backed the work all the way and has continued to do so ever since.

LETTER (continued)

The next step was to develop UNICORN, a tailor-made macro language which was less general than the AIDS package and thus much more economical to use. We also converted the existing data bases to Unicorn and developed the rudimentary basis for the steward's work pending his arrival. The concept of a steward as a 'system manager' was vital to this development as it allowed us to develop and implement the system cheaply and quickly as we did not have to foresee all possible eventualities and developments beforehand. With a steward a system can 'grow' and this was found to be a natural thing with a large system in APL. Hazel O'Hare and I did the UNICORN development in 6 weeks.

Roger Shaw, who was to be the system steward, then started work at the Wellcome foundation. He soon became familiar both with APL and the concepts we had used early in 1976. Accountants from the USA company and three European operating companies of Wellcome were initiated into the use of the system and, as access on the Sharp network became available, they were able to prepare their own plans on their own terminals. This meant far less drudgery than before. They were able to consider several alternatives and submit a 'best' plan saving about 2 weeks preparation time. The system is now used in Belgium, Holland, Germany, the USA and Canada and also in the UK operating company. For group purposes an almost 'perfect' (in accounting terms) consolidation is done at several levels which makes the final plan more comprehensive, realistic and valuable at all levels. Sensitivity analyses are done to evaluate, at the highest level, the effects of changing parameters, (for example exchange rates). 'What if' analyses are done at various input levels.

Since then several other systems have been developed, in particular to calculate profitability at product level based on forecast unit sales. The UNICORN system has grown and been upgraded many times and many people have been trained in the concepts and use of the system. For example, it has been upgraded to produce complex columnar output from several different data bases simultaneously, merely by setting a few simple state functions. Most of this work has been done by the Wellcome people together with Hazel and, latterly, David Weatherby of IPSA UK.

Technically, we kept breaking new frontiers in this work; one of the most interesting (though harrowing at the time), was transferring the data for the plan from the Wellcome's own computer tapes to APL files. Originally, we were given tapes from their DP Department's ICL machine, which we sent to Toronto by plane. We then discovered that it took about 5 days to get the tape to Toronto this way instead of the expected 24 hours. After battling our way through the red tape generated by the '3 C's' of couriers, carriers and customs officials, the first tapes eventually arrived in Toronto. They were pronounced unreadable! No one had a specification of how the tapes had been stored on the ICL equipment and, with no experience of this problem on hand we found it difficult to sort out until Roger Moore brought APL to the rescue. Roger, one of the original implementers of APL, had worked on ICL equipment several years before and recalled from memory the sequence of character codes used by ICL - and suddenly we could interpret them. Nowadays, that all seems like a faraway nightmare. Tapes are transmitted overnight to Toronto on the Mohawk equipment, and we've had lots of practice working with 'foreign' tapes so translating them is not a problem anymore.

We at IPSA have gone through quite a few new areas with our friends at the Wellcome, and I expect we'll find some new places to go again soon. To quote John White in his IFIP paper:

'I am now of the view that computers are nothing to do with computation, nor with information, but are all about communication at the highest level. If this view is correct we should be bracing ourselves for the industry being the motor force behind changes to society, comparable in scale to those caused by the invention of printing and the development of personal transport.'

Having lived through some fairly tremendous changes in the state-of-the-art since we met, I think he may well be right!

PUBLIC LIBRARY UPDATE

Changes in workspaces

39 <i>X11</i>	<i>HSPRINT</i> now replaces the old <i>PRINTREQ</i> functions.
39 <i>BOXJENKINS</i>	Model estimation significantly speeded up.
39 <i>MAGIC</i>	Currency and UKCSO data bases can now be accessed.
51 <i>FPSTOCKLIST</i>	A list of the five groups of securities available on the Financial Post.

Workspaces deleted

51 <i>AMFILE</i>	Redundant now that the package data type (which is cheaper) exists.
51 <i>CNFILE</i>	
51 <i>NYFILE</i>	
51 <i>OPTFILE</i>	

New workspaces

3 <i>GREDIT</i>	Graphics editor - see below.
5 <i>TERM</i>	See SATN-28.
45 <i>GENSIM</i>	Generalized Simulator - uses weight-directed graphs.
81 <i>CURRENCY</i>	New access functions to currency data bases.
86 <i>INSPECTOR</i>	See below.
92 <i>STARTREK</i>	Fun and games.

INSPECTOR

Mark Dempsey

INSPECTOR is a package for creating, maintaining, and accessing microfilm data bases. The original INSPECTOR system was developed for use on a minicomputer as a turnkey package. Recently, an APL version of the system was implemented, and it now resides in *LIBRARY* 86. The unique feature which distinguishes INSPECTOR from other data base systems is the fact that only part of the data is stored on-line, yet all data can be retrieved immediately via microfilm.

There are many possible applications for APL INSPECTOR. The first, a word processing system implemented for Eastman Kodak Co., is a system to file incoming and outgoing correspondence efficiently. Each user can microfilm documents as they pass through their department. Key information about the document is then entered into the SHARP APL File Subsystem, using INSPECTOR. Items such as the film's reel and frame number, the author and recipient of the document, the date, the type of document, and a list of keywords identify the subject matter. Once this data is entered, INSPECTOR can be used to retrieve documents on any one, or any logical combination, of these criteria. A Kodak microfilm viewer (which is hardwired to the APL terminal) automatically displays the requested documents in full.

This approach is ideal for recording data which is not easily stored in a computer using conventional means (i.e. drawings, blueprints, X-rays, schematics, etc.).

Inquiries about APL INSPECTOR should be directed to Mark Dempsey or Lael Kirk of the Rochester office. For information regarding the minicomputer implementation contact William Tate in the same office.

Technical Supplement - 12

FLOATING POINT ARITHMETIC

Doug Forkes

Multiplication is vexation
Division is as bad.
The rule of three doth puzzle me,
And fractions drive me mad.

Anon.

Floating point is a technique for representing large numbers and numbers with fractional parts. A number is represented as an integer with a fixed number of digits in conjunction with a second integer locating the decimal point. For example the number 12.345 which can be written 12345×10^{-3} could be stored as 12345, -3 This example illustrates five digits of precision in radix ten.

The precision of floating point arithmetic is governed by two parameters: the radix which is used and the number of significant digits employed. In SHARP APL on system 360 computers fourteen digits are kept in radix sixteen.

The purpose of this article is to explain why certain anomolous results occur when using fractional or large numbers in SHARP APL. Users who never use fractions or large numbers may omit this section.

Typical anomalies:

```

)CLEAR
CLEAR WS
+ / 2 1E60 -1E60 3 A 5 OF COURSE
3
1 | 1.2 x 10 A 0 RIGHT ?
1
)DIGITS 16
WAS 10
@*.5 A @* IS AN IDENTITY. ERGO .5
0.4999999999999999
7.5!8.5 A X!X+1 ↔ X+1
8.4999999999999999

```

These four examples demonstrate the four sources of error in floating point arithmetic.

1. Round-off error.

Although both 2 and $1E60$ can be represented exactly internally, their sum has too many significant digits to be stored exactly and must be truncated to $1E60$. Subsequently adding $-1E60$ and 3 gives the anomalous answer 3.

The astute student may be wondering why the reduction did not proceed right-to-left resulting in an anomalous result of 2. Very astute!

2. Wrong radix.

1.2 is a nice tractable terminating decimal in radix ten. Unfortunately in radix sixteen 1.2 is a repeating hexadecimal of 1.333... (hex). The hardware truncates this to 1.333333333333 (hex) which when multiplied by 10 yields 11.999999999999996 the 1 residue of which is 0.9999999999999996 which at `)DIGITS 10` SHARP APL rounds and prints as 1.

In groping for a solution to this problem the astute student (if he is still with us) may suggest that residue not return such an exact result. Perhaps, but surely it must be wrong to correct an error by making another one. That sort of thing could get out of hand.

3. Irrational numbers.

\sqrt{e} is the square root of e . No rational radix can represent this value. The result returned is therefore the natural log of the approximation to square root e . Of course that result would be irrational and it must be approximated. With luck, the approximation would be .5.

4. Imperfect algorithms.

Although 7.5, 8.5 and the result, 8.5 again, have exact internal representations, the algorithm which computes extended binary coefficients is not aware of this special case and cranks through many computations in which errors of types 1, 2, and 3 occur.

"The art is in concealing the art."

SHARP APL conceals the above errors by two techniques.

1. Rounding printed values to ten digits.

The normal `)DIGITS` setting is 10. This causes all floating point values to be rounded to ten digits when printed. Trailing zeroes are suppressed. Since most errors occur in the sixteenth significant decimal digit, results appear correct when printed. To illustrate this effect compare .07 printed at `)DIGITS 10` with `)DIGITS 16`.

2. Fuzz.

Basically fuzz is the concept of relative error. In computer languages which do not employ a fuzz, users must be warned to avoid straight forward comparisons of floating point values because the languages take those comparisons literally. In APL this approach is unacceptable for two reasons:

- a) APL tries to make programming easy. The user knows what he means when he types $A = .5$ and it is up to APL to adopt the same interpretation.
- b) APL does not require declarations of "type." Thus the user has no control over or indication of data type and therefore no way of distinguishing fixed point comparisons, which present no danger, from floating point ones which do.

"Money makes the world go 'round."

One frequently encounters misguided programmers who attempt to store financial data to the penny by means of floating point arithmetic (i.e. they record dollars to two decimal places of accuracy.) The result of some computation yields a fractional penny which must be rounded. For example round

$$\$14.5966 = \$12.34 \text{ (principal)} \times 1.18 \text{ (interest rate)}$$

The straight forward solution:

```

▽ R←ROUND R
[1] A ROUND TO 2 DECIMAL PLACES
[2] R←(⌊0.5+100×R)÷100
▽

```

works well for most R but runs into problems with $R = 12.374999\dots$. Ask some people to round this number to two decimal places and they will tell you 12.37 while others will insist that it is obviously the number 12.375 which is to be rounded, the result being 12.38. The function as written is sensitive to the number of trailing 9s, returning 12.37 until there are ten of them at which point it returns 12.38. The sensitivity arises from the fact that \lfloor is fuzzed. A solution avoiding the problem requires a second argument which is the number of decimal places of accuracy that are assumed to be present in the input.

```

▽ R←N ROUND R
[1] A ROUND FROM N DECIMALS TO 2
[2] R←⌊.5+R×10*N
[3] R←R+5×10*N-3
[4] R←(R-(10*N-2)⌊R)÷10*N
▽

```

Another source of error encountered by financial packages which attempt to store pennies as fractional dollars comes from subtraction. Subtraction of nearly equal values is the nemesis of floating point arithmetic. If error is present, subtraction can magnify it enormously. Compare

```

(1.2×1000000)=1200001-1
1
1=1200001-1.2×1000000
0

```

Users encountering this problem usually attempt to solve it by modifying fuzz ($\square CT$). Wrong! Unless you are very sure of what you're doing, $\square CT$ should be set to one of two values: its default of approximately $1E^{-14}$ or 0. In the above example the error originates in the subtraction of the two nearly equal values. It can be removed by avoiding floating point where possible

```
1=12000001-[1.2x1000000
```

1

That is good advice to APL system designers everywhere.

KNOCK-KNOCK

Roger Hui

The following can be used by security conscious systems to limit the time a user can spend in gaining access.

```

▽ KNOCKΔKNOCK;NTASK;PSWD;TTASK;Z
[1]  A PERMIT 60 SECONDS FOR ENTERING PASS-WORD
[2]  A <CLEAROUT> AND <NOCLEAR> FROM 1 WSFNS
[3]  CLEAROUT ◇ TTASK←(2 □WS 3)[10+□IO]
[4]  L00:→(0 1 ^.=xNTASK←□RUN ':: CONTINUE 2 61')ρL10
[5]  □DL 60
[6]  □BOUNCE TTASK ◇ →
[7]  L10:□ARBOU 0ρ□←'PASS-WORD? ' ◇ PSWD←□
[8]  →('OPEN SESAME'v.≠11↑PSWD)ρL10
[9]  Z←□BOUNCE NTASK[1+□IO] ◇ NOCLEAR
▽

```

The *TTASK* creates an *NTASK* whose sole purpose is to wait 60 seconds and then terminate the *TTASK*. If the correct password is given the *TTASK* within 60 seconds, the *NTASK* is terminated before it can issue the *TTASK*-terminating command.

As far as I know the function is foolproof. I am aware the 6I should be used instead of *CLEAROUT* and *NOCLEAR*, but using 6I would obscure the issue somewhat.

UTILITIES

Peter Teeson

Anyone who has programmed for any length of time soon discovers the need for a class of service programs. Indeed, most of us probably have a workspace with a collection of such functions. Examples are easy to give: copying a file or a portion thereof; general *GET* and *PUT* programs; vector to matrix, to name but a few of the more obvious cases.

This article will try to define utilities in general and then discuss two particular cases. Readers are invited to submit their own examples so that a public *UTILITIES* workspace can be created.

The first step, it seems to me, is to come up with a satisfactory definition of a "utility". I offer the following, and solicit improvements:

A program is a utility if:

- 1) It performs a general service.
- 2) It can be viewed as a "black box" whose contents can be altered with no distressing side effects.
- 3) It is used frequently.
- 4) It works in many different environments.

Two such services are *GET* and *PUT*. To what extent can we specify a general *GET* program?

In fact what we want to do is get some set of APL objects from a file. For example suppose we had a file that looked like this:

COMP. NO.	CONTENTS
1	15
2	⊠PACK 'A B C FOO'

where *A*, *B*, *C* are variables, and *FOO* is a well known function. Finally suppose that *C* is itself a package containing a compacted array (see NOTE 1).

So the specifications for *GET* might be

$\nabla R \leftarrow NL \text{ GET } FL \quad \text{aGET EXISTING NL OBJECTS FROM FILE } FL$

where *NL* is a name list (see NOTE 2) and

FL is: file tie number, component number, optional magic number.

If *NL* is elided assume all objects are desired.

The result should be either the value, for cases similar to component 1, or a name list of objects actually defined, in cases similar to component 2.

We should detect and 'uncompact' any compacted arrays mentioned in *NL*.

We should avoid as many *DOMAIN ERRORS* as reasonable when dealing with packages.

Finally the function should be fully restartable. It may also use other utilities where appropriate.

The comments from a version of such a function follow, and can also be found in component 1 of file 1142021 *UTILS*. The function itself is in component 2.

R←NL GET FL

*GET EXISTING NL OBJECTS FROM FILE FL.
 IF NOT PACKAGE RETURN WITH VALUE IN R.
 ELIDED LEFT ARGUMENT IMPLIES ALL OBJECTS.
 EMPTY NL MEANS WE'RE DONE NOW.
 ONLY EXISTING OBJECTS
 AND NON-SI FUNCTIONS
 GET DEFINED (TO PREVENT DOMAIN ERROR).
 WE NOW UNCOMPACT ANY COMPACTED ARRAYS.
 DONE? OR FUNCTION?
 UNPACK NEXT COMPACTED OBJECT.
 IN COMPACTED ARRAYS ONLY VARS ARE B,NB, BY DEFINITION.
 WHIRR.
 RETURN NAMES OF OBJECTS MATERIALIZED.*

Readers are invited to write their own versions before looking at the component 2 example, and to append them to the file as packages.

Note that the example uses two other utilities *VTM* and *DEC* and these will be discussed in the next article.

Finally you are invited to specify the *PUT* utility, appending the results to the same file.

Suggestions or comments on this article can be sent in the *MAILBOX* to *PHT* or can be appended to the above file.

Note 1

A compacted array is a mechanism for storing sparse arrays. For example suppose:

$A \leftarrow 100 \quad 10 \quad \rho \quad 0 \quad 1 \quad 0 \quad 2 \quad 0 \quad 3 \quad 0 \quad 4 \quad 0 \quad 5$

This array has a 'sparsity index' of 0.5 (the ratio of zeroes to number of elements). We may store this array as a package by doing the following:

$B \leftarrow A \neq 1 \uparrow 0 \rho A$
 $NB \leftarrow (,B) / ,A$
 $A \leftarrow \square \text{PACK } 'B \ NB'$

We now have a new object *A* that contains sufficient information to reconstruct the original *A*, but which consumes less space.

Reconstruction is achieved by:

$A \leftarrow (\rho B) \rho (,B) \backslash NB$

Thus we define a compacted array to be a package containing only two variables, *B* and *NB*, representing the original array.

Note 2

A name list is a scalar, vector or character array of names of APL objects. If a vector, the names are separated by blanks; if a matrix, one name appears in each row with optional leading and trailing blanks.

SHARP APL TECHNICAL NOTES

SATN-INDEX

15 SEP 77

SATN-0	1 JAN 76		SATN INTRODUCTION
SATN-1	1 JAN 76		TASKID
SATN-2	15 SEP 77	(Rev. 2)	CONTROL MESSAGES
SATN-3	1 JAN 76		□OUT
SATN-4	15 JUN 76	(Rev. 1)	NTASKs and BTASKs
SATN-5	15 JUL 76	(Rev. 1)	BATCH APL
SATN-6	1 JAN 76		EXECUTE
SATN-7	1 JAN 76		LATENT EXPRESSION
SATN-8	15 AUG 76	(Rev. 1)	HSPRINT
SATN-9	1 JUN 76		USAGE INQUIRY SYSTEM
SATN-10	1 JAN 77	(Rev. 1)	SORTREQ
SATN-11	1 JAN 76)RESET
SATN-12	1 JAN 76)COPY
SATN-14	15 JAN 77		PACKAGES
SATN-15	15 APR 76		INDEX
SATN-16	14 APR 76		FILE SYSTEM MUST-WRITE BUFFERS
SATN-17	30 JUN 76		FORMATTING PRIMITIVE
SATN-18	1 JUL 76		□FMT
SATN-19	1 JAN 77		FILEPRINT
SATN-20	15 JAN 77		SYSTEM VARIABLES AND FUNCTIONS
SATN-21	15 JAN 77		□WS and □FD
SATN-22	15 JAN 77		APL WORKSPACE TRANSFER
SATN-23	7 JUN 77		COMPARISON TOLERANCE
SATN-24	25 MAR 77)SYMBOLS
SATN-25	15 MAY 77		EXTENSIONS TO ARGUMENT PASSING
SATN-26	15 SEP 77		ENHANCEMENTS TO THE FILE SUBSYSTEM
SATN-27			
SATN-28	11 JUL 77		TERMINAL CONTROL

MAXICON CONNECTION - CONTEST RESULTS

Congratulations to all of those who took part in what has proven to be our most successful contest to date. The entrants and their submissions were:

Andy Baldwin	Statistics Canada	Ottawa	MAXIΔABA
Ashok Gupta	Xerox	Toronto	MAXIΔAGU1/2/4/5
Alfred Gilhausen	I.P.S.A.	Vancouver	MAXIΔAHG
Bernie Levine		Ottawa	MAXIΔBL
Clive Edwards	I.P.S.A.	Toronto	MAXIΔCED
Dinos Appa	I.P.S.A.	London	MAXIΔDINO
Dave Saunders	I.P.S.A.	London	MAXIΔDNS
Gar Buchwald	New England Mutual Life	Boston	MAXIΔGBU
Gerry Brown	Environment Canada	Ottawa	MAXIΔGBR
Grant McDorman	I.P.S.A.	Toronto	MAXIΔGMD
Geoff Lewis	Intersystems	Amsterdam	MAXIΔGRRL1/2
Clement Kent	I.P.S.A.	Toronto	MAXIΔKENT
Leon Mitchell	Aquitane Petroleum	Calgary	MAXIΔLM
Mark Dempsey	I.P.S.A.	Rochester	MAXIΔMDE
Oscar M. Nierstrasz		Toronto	MAXIΔOMN
Ron Bradley	I.P.S.A.	Toronto	MAXIΔRBR
Robert Hart	Duncan C. Fraser & Co.	Liverpool	MAXIΔRHA
Rohan Jayasekera	I.P.S.A.	Montreal	MAXIΔRHO
S.H. Cooper	Acadia Life	Toronto	MAXIΔSHC
Wolfgang E. Fendt	Stiftung Rehabilitation Ctr.	Heidelberg	MAXIΔWEF
Wayne Smith		Columbia, Md.	MAXIΔWSM
Yuen K. Wong	Treasury Board	Ottawa	MAXIΔYKW
Jon Leitch	Xerox	Toronto	MAXIΔJL
Irwin Pressman	Carleton University	Ottawa	MAXIΔISP

Fully 23 of the above 28 submissions were shown to be complete to within the stated contest specifications. Execution times were then measured as the 23 survivors were passed input matrices of varying complexity. A partial list of the ranked execution times expressed as a multiple of the fastest entry are shown below.

Function	Ranked Time
MAXIΔYKW	1.00
MAXIΔWEF	1.45
MAXIΔRBR	1.93
MAXIΔAHG	2.35
MAXIΔROH	2.49
MAXIΔBL	2.58
MAXIΔDNS	2.59
MAXIΔLM	3.62

Normalized execution times for the other submissions ranged from 4.67 to 64.09 with a mean normal execution time of 16.82. When the fastest of these entries was measured for minimality of internal storage requirements on large input matrices, there was virtually no difference in the overhead taken by MAXIΔYKW and MAXIΔWEF, while the storage required for the other entries was larger by factors ranging from about 1.4 to 4.7.

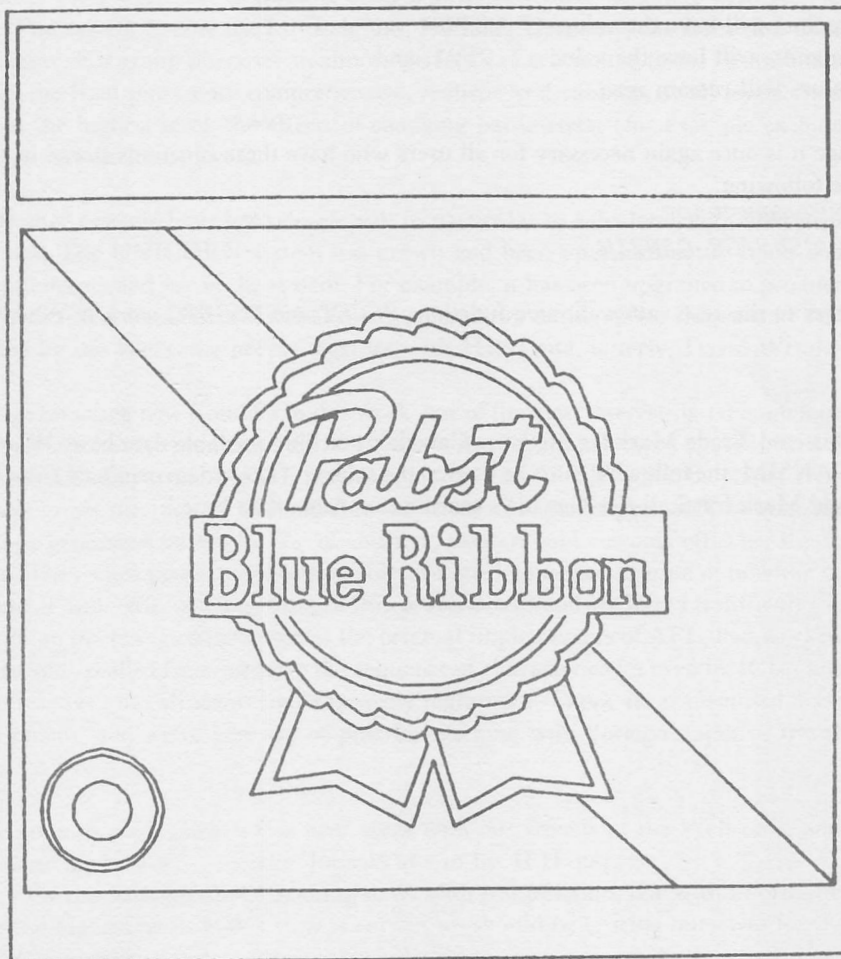
Accordingly, on the basis of speed and somewhat more concise code, our overall winner is ***Yuen K. Wong***, with a well-deserved honourable mention going to Wolfgang E. Fendt. The best I.P.S.A. entry belonged to Ron Bradley. Our winners will be contacted shortly and may look forward to receiving a book prize of their choice. For those interested, the winning entries may be viewed in workspace 999 CONTEST. MANYρTHANX\10 to all who participated.

Jerry Cudeck

GRAPHICS EDITOR

Dave King

Our new editor for graphics (in workspace 3 *GREDIT*) is designed for use on graphic terminals that support cross hair input. Several commands are available to enter a picture and to manipulate it. Simple, and fun; to get started, type *EDIT*. See the *DESCRIBE* in workspace 3 *GREDIT* for a description of commands.



CANSIM*

Lou Adams,
I.P. Sharp Associates, Ottawa.

As of October 1st, the new definition of the CANSIM MINI BASE will be available on the SHARP APL system. As mentioned in 81 *CSNEWS*, quite a few matrices have been deleted from the mini base. These will not be immediately deleted from the Sharp CANSIM MINI BASE but as of October 1st they will no longer be updated. To determine whether a particular series is affected, please check the 8th element of the 10 element string added to the front end of each series. If it has a non-zero value then it is a series which will be deleted. If there are any such series which you would like kept as part of the data base and updated, please contact your SHARP APL representative and arrangements will be made (at no cost).

Until this new release of the CANSIM MINI BASE, zero has been used to represent secure datapoints, padded datapoints and true zero datapoints. This is still the case if the function *CSGET* is used to retrieve the time series. If the function *CSGETC* is used (with the same format as *CSGET*) then:

secure datapoints will have the value $-7.2E71$
padded datapoints will have the value $7.2E71$, and
true zero values will remain zero.

Because of this change it is once again necessary for all users who have these functions stored in their own private workspaces to do the following:

```
)LOAD YOURWORKSPACE  
)COPY 81 CSUSAGE CANSIM  
)SAVE
```

Other than the changes to the zero values the two functions *CSGET* and *CSGETC* work in exactly the same way.

*CANSIM is the registered Trade Mark for Statistics Canada's machine readable data base. When publishing any data retrieved from CANSIM, the following must be used as the source: "These data originate from CANSIM which is the registered Trade Mark for Statistics Canada's machine readable data base."

SHARP APL COURSES

Introduction to APL:	OCT	NOV	DEC	JAN	FEB
CALGARY (5 days, over 4 wks:	12,13,19,26,Nov.2				
DALLAS	17-21				
HOUSTON	14-18				
OTTAWA	10-14	14-18	12-16	16-20	
ROCHESTER	17-21	21-25	12-16	16-20	
SEATTLE	31 →	01-02			
TORONTO	11-13		12-14	04-06	15-17
TORONTO	01 →	01-02		25-27	
U.K. (LONDON)	17-19	21-23			
VANCOUVER	26-28	30	→ 2	11-13	08-10
WINNIPEG		21-24		23-26	20-23

Intermediate

TORONTO	On demand
SEATTLE	Nov.10-11
WINNIPEG	On demand

Advanced

U.K.	"APL System Design"	Oct. 24-25
------	---------------------	------------

Seminars

EDMONTON:	"SHARP APL Public Data Bases "	Dec. 15
	"Good APL Programming Techniques"	Jan. 19
TORONTO:	"MAGIC for Time Series Analysis"	Nov. 14
U.K.:	"MAGIC for Time Series Analysis"	One-day courses,
	"Appreciaton of APL"	arranged on demand.
	"Introduction to the SHARP APL system"	
VANCOUVER:	"APL User Seminars"	Monthly from October.

Munich Messe - SYSTEMS 77

APL Demonstrations will take place from October 17 to 21.

Look for the I.P. Sharp Associates stand - we hope to see you there.

Sharp Special Systems

DISTRIBUTED PROCESSING OR DECENTRALIZED CHAOS?

Hugh O'Rourke

One characteristic of the computer industry which has made it so fascinating to work in over the years is that, so far as technology is concerned, it is what might be called 'leading edge.' The inference here is not that everyone in the industry works all the time with the latest gizmos, but rather that the technology has advanced faster than our ability to either understand or properly use it. This is most obviously true with respect to hardware, but it applies to software as well.

Take, for example, minicomputers, the first of which was produced twenty years ago. It is not uncommon (even, would you believe it, at IFIP '77) to hear statements such as 'in the ten years since the first minicomputer was introduced' Well, there is an element of truth in this, because it is certainly only within the last ten years that system-level software has been available to make minis really usable.

One would expect, however, a protest from some of those pre-'67 minis that are now so old as to be suspected of senility. In fact, if you'll pardon the digression, I don't think people know what to do with old minis. Unlike automobiles, they do not have graveyards, and although it has been suggested, none have yet been bronzed to hang in the lobby.

The fact that minicomputer software lagged so far behind the hardware is a good example of how one area of the industry outdistanced another, with reasonably disastrous - in a financial sense - consequences to the customer. This is, in fact, one of the problems facing distributed processing. Although there are, for example, good database management packages on the market now, few if any support a distributed data base.

The situation with the greatest potential for disaster, however, arises when the technology of the industry as a whole (whether real or just advertised) outdistances the ability of the users of the products of the industry to either understand or properly use them. This is surely the biggest hurdle to distributed processing.

There is not yet a single name or standard definition of the concept. Perhaps this is inherent in the very nature of the whole idea which is the functional distribution of data processing activities along logical organizational lines. This implies of course, that a distributed system can be implemented to parallel the flow of work and data within any organization's structure.

DISTRIBUTED PROCESSING (continued)

The potential benefits of distributing data processing are enormous. Distributing a workload over several processors generally costs less than a centralized system. In relation, say, to a centralized system with remote terminals there will be savings in communications costs, if the data base is logically distributed along with the processing. Expansion is both less expensive and more flexible. Backup can be achieved with far less duplication and downtime becomes a matter of reduced capability or degraded performance rather than a total loss of processing.

But distributed processing also presents an enormous challenge to management if these potential benefits are to be realized. The danger of distributing a data base is that you might lose control of it. A great deal of planning will be required to minimize the redundancy of data. Although distributed systems can be more 'friendly' to users in the sense that a local portion of the system can be more immediately responsive to its users' needs, the trick will be to distribute just enough control to achieve this but not so much as to permit an unnecessary proliferation of technical personnel. Do you let local personnel develop any of their own software, and if you do, how do you ensure that the resulting local systems will be both hardware and software compatible with the rest of the systems in the network?

To make a distributed system really work and be cost-effective, it must be designed from the top down and be implemented from the bottom up. This is not the way systems have been built in the past, and management must become more directly involved if they expect the resulting system to satisfy both the needs of each functional unit within their companies and provide them (the managers) with good access to the data they require - not only to run the company day-to-day, but also to plan for the future.

So far, distributed processing has been more talk than anything else. Aside from the fact that the onus is on management to ensure that distributed systems don't turn into the same sort of incomprehensible monsters that some of the larger centralized ones have, the computing industry has yet to provide all of the tools to really make it work.

But when distributed processing does finally come of age, so too in a sense will this whole business of automated processing and storing of data. For distributed processing, in its ultimate form, offers the potential for 'closing the loop' with respect to the total flow of work and data throughout any organization. And when this happens, the result will be not only an efficient, cost-effective, integrated and organization-wide processor of data, but as well be a true management information system.



INTERNATIONAL BRANCH OFFICES

Amsterdam

Intersystems B.V.
Herengracht 244
1016 BT Amsterdam
The Netherlands
(020) 24 40 50

Birmingham

I.P. Sharp Associates Limited
2nd Floor, Radio House
79/81 Aston Rd. North
Birmingham B6 4BX
England
021-359-964

Boston

I.P. Sharp Associates, Inc.
Suite 812
148 State St.
Boston, Mass. 02109
(617) 523-2506

Brussels

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

Calgary

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

Chicago

I.P. Sharp Associates, Inc.
Suite 424
8501 West Higgins Rd.
Chicago, Ill. 60631
(312) 693-5895

Copenhagen

I.P. Sharp ApS
Østergade 24B
1100 Copenhagen K
Denmark
(01) 112 434

Dallas

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

Düsseldorf

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

Edmonton

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

Gloucester

I.P. Sharp Associates Limited
29 Northgate St.
Gloucester, England
0452 28106

Houston

I.P. Sharp Associates, Inc.
Suite 405, One Corporate Square
2600 Southwest Freeway
Houston, Texas 77098
(713) 526-5275

London, Canada

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

London, England

I.P. Sharp Associates Limited
132 Buckingham Palace Rd.
London SW1W 9SA
England
(01) 730-0361

Minneapolis

I.P. Sharp Associates, Inc.
Suite 1371, 1 Appletree Square
Bloomington, Minn. 55420
(612) 854-3405

Montreal

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

New York City

I.P. Sharp Associates, Inc.
Suite 250, East Mezz.
Pan Am Bldg.
New York, N.Y. 10017
(212) 986-3366

Newport Beach

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

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 110, 299 California Ave.
Palo Alto, Ca. 94306
(415) 327-1700

Rochester

I.P. Sharp Associates, Inc.
Suite 1150, 183 Main St. E.
Rochester, N.Y. 14604
(716) 546-7270

San Francisco

I.P. Sharp Associates, Inc.
Suite C409, 900 North Point St.
San Francisco, Ca. 94109
(415) 673-4930

Seattle

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

Stockholm

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

Toronto

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

Vancouver

I.P. Sharp Associates Limited
Suite 604, 1112 West Pender St.
Vancouver, B.C. V6E 2S1
(604) 682-7158

Victoria

I.P. Sharp Associates Limited
870 Roy Road
Victoria, B.C. V8Z 2X3
(604) 479-1611

Vienna

I.P. Sharp Ges. MbH
Rechte Wienzeile 5/II
3 Wien, Austria

Warrington

I.P. Sharp Associates Limited
48A Horsemarket Street
Warrington, Cheshire
England
55342

Washington

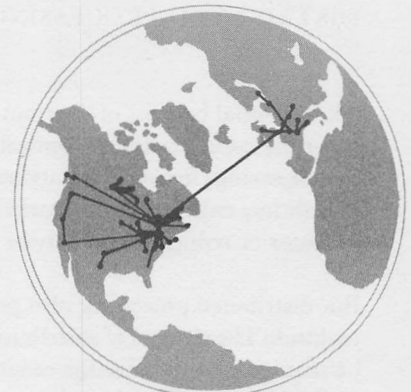
I.P. Sharp Associates, Inc.
Suite 307, 1730 K Street N.W.
Washington, D.C. 20006
(202) 293-2915

Winnipeg

I.P. Sharp Associates Limited
Suite 909, 213 Notre Dame Ave.
Winnipeg, Manitoba R3B 1N3
(204) 947-1241

Zürich

I.P. Sharp A.G.
Badenerstrasse 141
8004 Zürich
Switzerland
241 52 42



APL Operator Voice (416) 363-2051

SHARP APL Local Access In:

Canada

Calgary
Edmonton
Halifax
Hamilton
Kitchener
London
Montreal
Ottawa
Quebec City
Regina
Saskatoon
Toronto
Vancouver
Victoria
Winnipeg

U.S.A.

Ann Arbor
Boston
Buffalo
Chicago
Dallas
Dayton
Des Moines
Houston
Los Angeles
Minneapolis
Newport Beach
New York City
Rochester, N.Y.
San Francisco
Seattle
Stamford, Ct.
Syracuse, N.Y.
Washington, D.C.
White Plains

Europe

Amsterdam
Bruxelles
Copenhagen
Coventry
Düsseldorf
Gloucester
Liverpool
London
Manchester
Milan
Paris
Stockholm
Vienna
Zürich

TELENET AND TYMNET

Alabama

Birmingham

Arizona

Phoenix

California

Alhambra
El Segundo
L.A. Central & East
L.A. West & Valley
L.A. South & Long Beach
Mountain View
Oakland
Orange County
Oxnard

Riverside

Sacramento
San Clemente
San Carlos
San Diego
S.F. Peninsula
San Jose
Santa Ana
Santa Barbara
Santa Rosa

Colorado

Denver
Colorado Springs

Connecticut

Bridgeport
Danbury
Darien
Hartford
New Haven/Bridgeport
Waterbury

Delaware

Wilmington

Florida

Fort Lauderdale
Jacksonville
Miami
Orlando
Pensacola
St. Petersburg
Tampa

Georgia

Atlanta

Idaho

Boise

Illinois

Freeport
Rockford
Indianapolis
South Bend
Iowa
Cedar Rapids
Iowa City
Kansas
Wichita

Kentucky

Louisville

Louisiana

Baton Rouge
Lafayette
New Orleans

Maryland

Baltimore

Massachusetts

Cambridge

Michigan

Detroit
Jackson
Kalamazoo
Southfield

Missouri

Kansas City
St. Louis

Nevada

Las Vegas

New Hampshire

Nashua

New Jersey

Englewood Cliffs
Moorestown
Princeton
Piscataway
Newark
Union
Wayne

New York

Corning
Hempstead
Long Island
Niagara Falls

North Carolina

Durham
Raleigh/Durham

Ohio

Akron
Cincinnati
Cleveland
Columbus
Toledo

Oklahoma

Oklahoma City

Oregon

Portland

Pennsylvania

Allentown
Erie
Harrisburgh
Philadelphia
Pittsburgh
Valley Forge
York

Rhode Island

Providence

Tennessee

Memphis

Texas

Austin
Baytown
Beaumont
El Paso
Ft. Worth
Midland
Odessa
San Antonio

Utah

Salt Lake City

Virginia

Norfolk

Wisconsin

Madison

Milwaukee

Oshkosh

- ☐ Please amend my mailing address as indicated.
- ☐ Add to your mailing list the following name(s).
- ☐ Send me a SHARP APL publications order form.

Name: _____

Co.: _____

Address: _____