

# ↑ PRTSC

Print SScreen: The newsletter of the Stanford/Palo Alto Users' Group for the IBM PC

PO Box 3738

Stanford, CA 94305

Volume 2 Number 5

May 1984

```
*****
* Next meeting:      Wednesday,      *
*                   7:00pm           *
*                   Rm. 111 Polya Hall *
*                   Stanford University *
*                   *                   *
*****
```

## AGENDA FOR THE MEETING --

7:00 General Club Business

Report on the Monday before the Wednesday Meeting  
New Business

7:30 Reports

8:30 Random Access

9:00 Conclusion of the Official Club Meeting

(Don't forget the Monday before the Wednesday meeting at Talbott's -  
8:00 P.M. - 463 California Ave. - Come join us with suggestions/comments)

## CLUB OFFICERS

Treasurer: Beverly Altman 329-8252  
Speaker's Bureau: Mike Van Waas 325-2507  
Soft Copy Librarian: Jim Caldwell 692-7181  
Hard Copy Librarian: Joe Wible 497-6585  
Bulk Purchasing: Greg Tinfow 493-7404  
Membership: Linda de Sosa 856-6281  
Bulleting Board: Lindsey Puckett 968-1809  
Newsletter: Jackie Carr 858-1641  
Periodical Review: Kevin Ohlson 494-2574

**MEETINGS:** Last Wednesday of each month - 7 PM

## MEMBERSHIP FEE:

\$25.00 (includes a club library listing  
diskette which you sign out & return)  
1983 newsletters are included.

**NEWSLETTER:** Deadline for submitting  
articles, comments, or information -  
10 days before each meeting.

**CLUB BULLETIN BOARD** 968-1809  
6 PM to 11 PM

## SPECIAL INTEREST GROUPS:

DataBase - Greg Tinfow 493-7404  
Communications - Glen Ingram 323-1669  
Business - Linda de Sosa 856-6281  
Expansion Boards - Curt Carlson 941-5680

## VOLUNTEER RESOURCE PEOPLE:

Tony Nunas (857-0836) - PC Write  
Jackie Carr (858-1641) - Word Star  
Les Weil (321-5541) - Freeware  
(Copies By Appointment)  
Jeff (Days: 321-5930) Pascal; Dos 2.0  
Paul 968-8283 - APL  
Jim Webster (326-3365) - Multiplan  
Corwin Nichols (494-8640) - "C";  
Floppy Disk Formats  
Greg Tinfow (493-7404) - AST Boards

## ADVERTISING RATES

\$25.00 - 8.5" x 11" (full page)  
\$15.00 - 8.5" x 5.5" (half page)  
\$10.00 - 4.2" x 5.5" (quarter page)

**QUESTION:** I have a computer at my office and one at home, but they're not compatible. Is there an easy way to transfer data between them?

**ANSWER:** The easiest way is to do a little primitive networking. Use an information utilities as a bridge. Buy modems for both computers. When you leave the office, store the data you want in the information (BB) mailbox. When you get home, call the Bulletin Board with your home computer and pick up the data. (Data transfer is easiest if similar programs are run in both places: Visicalc, WordStar, etc. It's cheaper to subscribe to an information utility like The Source and buy a couple of modems than it is to get another computer that's compatible)

**DESQ:** A software integrator lets you run several programs from different manufacturers on your video screen at the same time in individual "windows". You can run Lotus 1-2-3-, WordStar, dBASE II and custom-designed programs plus communications packages. You can transfer information between windows (word-processing to a graph or spreadsheet or transfer data from one database to another) \$399.

### OPERATING SYSTEMS FOR THE IBM-PC

<b>Operating System</b>	<b>Vendor</b>	<b>Price</b>	<b>Type</b>	<b>Features/Comments</b>
MS-DOS 2.0	Microsoft Corp. Bellevue, Wash.	\$ 60	single user, single tasking	some compatibility with Xenix
CP/M-86	Digital Research, Inc. Pacific Grove, Calif.	\$ 60	single user, single tasking	compatible with CP/M-80 files
Concurrent CP/M-86	Digital Research, Inc.	\$ 350	single user, multitasking	windows; program compatibility with CP/M-86; file compatibility with CP/M-80; runs MS-DOS applications
MP/M	Digital Research, Inc.	\$ 650	multi-user	not currently sold in IBM PC disk format; can be obtained in 8-inch IBM 3740 disk format; will eventually be replaced with multi-user Concurrent CP/M
Venix	published by Venturcom Cambridge, Mass. PC version distributed by Unisource Software Corp. Cambridge, Mass.	\$ 800—single-user system \$1000—multi-user system	single or multi-user	licensed Version Seven Unix with Berkeley enhancements, C compiler; uses only 3.5 megabytes of hard-disk capacity
Coherent	Mark Williams Co. Chicago, Ill.	\$ 500	multi-user	runs Unix software; runs MS-DOS 2.0 software
Xenix	Microsoft Corp. through The Santa Cruz Operation Santa Cruz, Calif.	\$ 595—Package 1 \$ 595—Package 2 \$ 495—Package 3 \$1350—all three	multi-user, multitasking	Module 1 is basic operating system; Module 2 is programming tools; Module 3 includes word processing, other applications
Pick	Pick Systems Irvine, Calif.	\$ 495	multi-user, multitasking	has built-in data base; allows sharing of PC-DOS files; can download tasks from minicomputers and mainframes running Pick
Oasis	Phase One Systems, Inc. Oakland, Calif.	\$1495	multi-user, multitasking	has electronic mail facility, print spooling; maintains user log
Turbodos	Software 2000, Inc. Arroyo Grande, Calif.	\$ 250—single user \$ 750—multi-user	multi-user, multitasking	links both 8- and 16-bit personal computers into networks
PC/IX	IBM Corp Boca Raton, Fla.	\$ 900	multi-user, multitasking	System III Unix; will exchange files with PC-DOS 2.0

## What or Why APL?

APL is a unique programming language, quite unlike most others. It bears no resemblance to FORTRAN, Basic, Pascal, etc. It does resemble in some way FORTH, and these two languages are in a unique class, quite distinct from the others. If you have tried FORTH, looked interesting but awkward to use for an occasional job, perhaps APL is the language for you.

APL is very interactive. In a certain sense, it is like using a programmable calculator with functions extended far beyond arithmetic into an incredible set of logic and sorting operators.

The language is arranged so that the overhead (dirty business) of implementing various operations is handled in the language. For example, looping and dimensioning statements are not needed in APL. They can be used however, if you wish.

this is not a compact language: IBM's APL interpreter occupies almost 70 Kbytes. A minimum of 128 K is required for either the IBM or STSC versions. I ran out of memory once with 128K, and was I surprised! It chews up memory quickly. Here are a few examples of the way APL works.

3 x 4

12

Note that no "print" command is necessary.

2 \* 3

8

Exponentiation.

9 - 2 x 5 - 3

5

9 - (2 x 5) - 3

2

There is no hierarchy of operations as in Basic. Expressions are evaluated and execute from right to left, modified by parentheses.

4 7 3 9 + 1 6 5 0.2

5 13 8 9.2

Elements of the object on the left (a vector) are added to the object on the right without need for dimensioning or looping. They must be conformal, that is, the same number of elements in each object.

These basic primitive operators can be naturally extended, in a comfortable, top down, structured way. A user function is written as follows:

▽ C ← TEMP F

[1] C ← (F-32) x 5 + 9

▽

and used in the following way:

TEMP 68

20

Some of these examples were taken from:

"APL: The Language and its Usage" by R. Povlika and S. Pakin  
"STSC APL\* and IBM PC APL: Two APLs for the IBM PC" by J. Bensimon, Byte magazine, March, 1984

## What or Why APL?

APL is a unique programming language, quite unlike most others. It bears no resemblance to FORTRAN, Basic, Pascal, etc. It does resemble in some way FORTH, and these two languages are in a unique class, quite distinct from the others. If you have tried FORTH, looked interesting but awkward to use for an occasional job, perhaps APL is the language for you.

APL is very interactive. In a certain sense, it is like using a programmable calculator with functions extended far beyond arithmetic into an incredible set of logic and sorting operators.

The language is arranged so that the overhead (dirty business) of implementing various operations is handled in the language. For example, looping and dimensioning statements are not needed in APL. They can be used however, if you wish.

this is not a compact language: IBM's APL interpreter occupies almost 70 Kbytes. A minimum of 128 K is required for either the IBM or STSC versions. I ran out of memory once with 128K, and was I surprised! It chews up memory quickly. Here are a few examples of the way APL works.

3 x 4

12

Note that no "print" command is necessary.

2 \* 3

8

Exponentiation.

9 - 2 x 5 - 3

5

9 - (2 x 5) - 3

2

There is no hierarchy of operations as in Basic. Expressions are evaluated and execute from right to left, modified by parentheses.

4 7 3 9 + 1 6 5 0.2

5 13 8 9.2

Elements of the object on the left (a vector) are added to the object on the left without need for dimensioning or looping. They must be conformal, that is, the same number of elements in each object.

These basic primitive operators can be naturally extended, in a comfortable, top down, structured way. A user function is written as follows:

▽ C ← TEMP F

[1] C ← (F-32) x 5 ÷ 9

▽

and used in the following way:

TEMP 68

20

Some of these examples were taken from:

"APL: The Language and its Usage" by R. Povlika and S. Fakin

"STSC APL\* and IBM PC APL: Two APLs for the IBM PC" by J. Bensimon, Byte magazine, March, 1984