Letters to the Editor

Change *CACM* and *Journal* Format?

EDITOR:

Many times I have wondered if ACM publications could be organised and printed in such a way to suit the needs of some readers such as myself. There is no doubt that the quality of the publications is excellent. However, it is impossible to keep all except articles of interest. The organisation and printing of the articles nevertheless do not permit independent filing. Because of this, may I propose the following rules for your consideration:

(1) To classify all articles by using the same classification scheme as adopted in your *Computing Reviews* by printing the classification code, say, on the top righthand corner of the page.

(2) All articles should start on an odd numbered page.

(3) No part of the article should share space with another article on the same page or at the back of that page. The space which becomes available could be used more effectively for advertising or contemporary news etc. Paradoxically, the advertising would be more effective as it would be noticed more frequently rather than be segregated so uniformly that the reader automatically knows how to avoid looking at it.

(4) Pages should be easily torn out and have a wide enough lefthand margin in order to punch holes.

I am thoroughly convinced that in the age of information explosion, and with the limited availability of space for storage, the adoption of similar rules to achieve the objectives desired would be of very great benefit to all your members.

A. GABAY
Wimbledon
London S. W. 20 England

Comments on Time Sharing

EDITOR:

The paper "Time-Sharing on a Computer with a Small Memory," by R. O. Fisher and C. D. Shephard, published in the February 1967 issue of *Communications of the ACM*, somewhat misrepresents my position on the memory requirements on time sharing.

It states that I thought a million words was required for effective time sharing, whereas this was what I thought was required for effectively meeting the needs of the whole MIT community at that time. In fact, I agree with the authors that it is possible to have effective time sharing systems on computers with small memories and I would like to call their attention to the paper "Time-Sharing Debugging System for a Small Computer," by McCarthy, Boilen, Fredkin, and Licklider, in SJCC 1963 and "THOR—A Display Based Time-Sharing System," by McCarthy, Brian, Feldman and Allen, in SJCC 1967. Both of these papers describe time sharing systems for computers with small memories.

JOHN MCCARTHY
Stanford University
Stanford, California 94305

Another Aspect of Economical Polynomials*

EDITOR:

In his paper "Methods of Evaluating Polynomial Approximations in Function Evaluation Routines" [*Comm. ACM 10*, (March 1967)], C. T. Fike fails to discuss one very important aspect of the "economical" methods for polynomials. Since these evaluation methods involve a decreased number of arithmetic operations over the usual Horner's method (or at least replace a multiplication by an addition) the implication is that they are faster to execute. Dr. Fike points out that these methods can be poorly conditioned for particular polynomials, thus requiring extended precision or fixed-point arithmetic to maintain accuracy and costing more in time than Horner's method. But even if we assume the methods are well conditioned, the need to store away and retrieve intermediate results in some machines with only one floating-point arithmetic register can wipe out the time savings effected by a reduction in the number of arithmetic operations. On many of today's high-performance computers the time required to store away and retrieve a result is about the same as the time required for a floating-point addition. It is no longer sufficient to estimate the efficiency of a method by a count of arithmetic operations alone.

To illustrate this point I have compiled the following table of timings for the "economical" methods vs. the usual Horner's method for several different machines (based upon instruction execution times published in the machine manuals). The assumptions are that the argument x is stored in memory on the Control Data 3600 and in a floating-point register on the System/360 machines. Horner's method is implemented without looping.

Degree of Polynomial	Method	Execution times in μsec CDC 3600	IBM 360/50	IBM 360/75
4	Horner	44.60	113.52	12.48
4	"Economical"	51.97	98.90	11.23
5	Horner	55.25	141.15	15.50
5	"Economical"	58.37	122.65	13.80
6	Horner	65.90	168.78	18.52
6	"Economical"	72.63	135.66	15.57

These "economical" techniques, at least, *are not economical* on the CDC 3600. They save time on the System/360 machines primarily because of the presence of extra floating-point arithmetic registers.

REFERENCES:

1. Control Data 3600 Computer System Reference Manual. Control Data Publication 6002 1300, 1964.
2. System/360 Instruction Timing Information. IBM Sys. Ref. Library, A22-6825-1, 1964.
3. IBM System/360 Model 75 Functional Characteristics. IBM Sys. Reference Library A22-6889-0 as revised by TNL N22-0209-0, 1965.

WILLIAM J. CODY, JR.
Argonne National Laboratory
Argonne, Illinois, 60439

*This work was performed under the auspices of the Atomic Energy Commission.

(*Letters are continued on page 537*)

At the moment, American society is barely entering the beginning stage of this debate over data surveillance. We can see that three quite different approaches are already appearing. One position, reflected by the initial views of many newspaper editors, civil liberties groups and congressional spokesmen is to oppose creation of data centers and intelligence systems completely. The need for better statistics for policy analysis or of richer information systems for criminal justice purposes is seen as inadequate when weighed against the increase in government power and fears of invasion of privacy that such systems might bring.

A second view, reflected in the initial thinking of many executive agency officials and computer scientists assumes that traditional administrative and legal safeguards, plus the expected self-restraint of those who would manage such systems is enough to protect the citizen's privacy. The more reflective spokesmen in this group would add that a large-scale decrease in the kind of personal privacy we have through inefficiency of information collection may well be on its way out, but that this would be something individuals could adjust to and would not seriously threaten the operations of a democratic society.

The third position, which I have tried to describe in my earlier discussion, assumes that neither the "total ban" nor the "traditional restraints" positions represent desirable alternatives. What is called for is a new legal approach to the processing of personal information by authorities in a free society and a new set of legal, administrative, and system protections to accomplish this objective. The fact is that American society wants both better information analysis *and* privacy. Ever since the Constitution was written, our efforts to have both order and liberty have succeeded because we found ways to grant authority to government but to tie it down with the clear standards, operating procedures and review mechanisms that protected individual rights. A free society should not have to choose between more rational use of authority and personal privacy if our talents for democratic government are brought to bear on the task. The most precious commodity we have now is the few years of lead-time before this problem grows beyond our capacity for control. If we act now, and act wisely, we can balance the conflicting demands in the area of data surveillance in this same tradition of democratic, rational solutions.

LETTERS—Continued from p. 531

Precision Calculations of e and π Constants

EDITOR:

The mathematical constants e and π have been calculated to absurd precision for some years now [1-3]. From a computational point of view much of this work is of little value, yet there are still some little unsolved problems or annoyances, which remain to plague designers of ultra-high precision scientific computational systems.

The difficulty arises because assemblers and compilers are hardly ever designed to convert decimal constants to a precision of more than a dozen or so digits. Thus, if calculations to greater precision are to be done, constants usually must be input in octal or other binary-derived representation. However, with the exception of the National Bureau of Standards' *Handbook of Mathematical Functions*, which on page 1017 gives e, π and some other related constants to 12 octal digits, publications of these constants in bases other than ten are hard to find. To correct this shortcoming, below are presented the values of the two constants in decimal, octal, and hexadecimal, each correct to 100 digits. I trust this precision will suffice for a number of years: I can't imagine the utility of further precision.

REFERENCES:

1. SHANKS, D., AND WRENCH, J. W., JR. Calculation of π to 100,000 decimals. *Math. Comput. 16* (1962), 76.
2. WHEELER, D. J. The calculation of 60,000 digits of e by the Illiac. Internal Report No. 43, Digital Comput. Lab., U. of Illinois, Urbana, Ill., 1953.
3. REITWIESNER, G. An ENIAC determination of π and e to more than 2000 decimal places. *MTAC 4* (1950), 11-15.

LYNN YARBROUGH
Arcon Corporation
Wakefield, Mass. 01880

					π					
Decimal	3.14159	26535	89793	23846	26433	83279	50288	41971	69399	37510
	58209	74944	59230	78164	06286	20899	86280	34825	34211	70679
Octal	3.11037	55242	10264	30215	14230	63050	56006	70163	21122	01116
	02105	14763	07200	20273	72461	66116	33104	50512	02074	61615
Hexadecimal	3.243F6	A8885	A308D	31319	8A2E0	37073	44A40	93822	299F3	1D008
	2EFA9	8EC4E	6C894	52821	E638D	01377	BE546	6CF34	E90C6	CC0AC
					e					
Decimal	2.71828	18284	59045	23536	02874	71352	66249	77572	47093	69995
	95749	66967	62772	40766	30353	54759	45713	82178	52516	64274
Octal	2.55760	52130	50535	51246	52773	42542	00471	72363	61661	34705
	40747	05515	51265	17023	31010	50620	63767	46223	47347	04446
Hexadecimal	2.B7E15	1628A	ED2A6	ABF71	58809	CF4F3	C762E	7160F	38B4D	A56A7
	84D90	45190	CFEF3	24E77	38926	CFBE5	F4BF8	D8D8C	31D76	3DA06