===

Recreational & Educational Computing Newsletter

===

July/Aug. 1986 - Volume 1, Number 4.

Editor & Publisher: Dr. Michael W. Ecker. Executive Assistant: Susan Ecker.
Published six times per year at 129 Carol Drive/ Clarks Summit, PA 18411.
Publication date is the first week after the two months of the issue date.
Annual subscription $18.95; sample and back issues $3.50.
Reproduction rights $25 extra. Editor's chat line: (717) 586-2784.

Editorial

Welcome to our summer catchup issue! We have lots of goodies in store, including Part 3 of Strange Attractors and Black Holes, with an explanation (of sorts) of last month's card trick, and other matter. In upcoming issues, I'll have some card tricks just for fun, with some mathematical explanation of them as well. Sometimes they won't fit perfectly into computerization themes, but fun is okay for its own sake, isn't it?

As part of our catchup material, we have solutions to the Bingo question involving an N-game from issue #2, as well as from at least one question from issue #3.

Our sweepstakes and contest have attracted some attention in the computer press. The Computer Shopper, for which I'm a contributor, gave REC a free half page release in the Aug. 1986 issue, generating a number of inquiries and a subscriber or two. Details of the sweeps and the contest appeared on page 4 of last issue of REC (#3), and many of you subscribers have already entered both portions. Some of the solutions to the contest portion have proven to be quite good, and you readers continue to make me feel humble, indeed, with your impressive knowledge and skills. But more of you should be entering! There is no reason not to enter at least the sweepstakes portion. You can do this easily and automatically, too. How? Simple: Just send in your subscription extension now. Bear in mind that the more subscribers, the more prizes REC will be giving away on Feb. 2, 1987. (Nobody asked, but that's my 37th birthday.)

In other news, your editor evidently will not be joining the ranks of the fully self-employed so soon, for I've just been offered a temporary professorship at a Pennsylvania State University branch campus (Wilkes-Barre). Part of the deal: a three-day per week teaching schedule, allowing me time for all of you.

Enough talk! Let's get right to the goodies inside. But remember: I want to hear from more of you! Enjoy!

Mike

Michael W. Ecker, PhD; Editor/Publisher
Recreational & Educational Computing Newsletter

Philosophical and Practical Implications of Self-Reference

In issue #1 of REC, we explored the idea of self-reference from the viewpoint of art, recreations & mind-games, and programming. The last element culminated with a program in Basic which produced the exact same result, regardless of whether listed or run, without use of the list command, use of input/output, data files, etc.

For those who would say that such recreations are useless, consider the practical application in software design. For instance, if you word process, you know that a certain key is usually designated a command key. Quite a few years ago, Scripsit for the TRS-80 was the first word processor I used, and I recall that the @ key was the command key. Thus, pressing this key and the letter S did not produce the symbols @S on the screen, but rather instructed to insert. But what if I really wanted to print out the @ symbol?

Well, you then had to press some other combination. And so on. This is part of the reason for the creation of special function keys, keys used for editing or manipulating, rather than for the actual input. There are numerous other instances of this in other word processors, including the one I'm using to produce REC now.

For those of you who prefer that we steer away from the educational and back to the recreational, you might enjoy this little challenge of a related nature. You have n sentences, where n is a positive integer (positive whole number):

Sentence 1: The number of false sentences in this list is 1.
Sentence 2: The number of false sentences in this list is 2.

.... (and so on for a total of n sentences)

Sentence n: The number of false sentences in this list is n.

Question: Which sentences are true? Which are false? (Hint: If you are stuck, then stick to a specific value of n, say 4.)

This issue of REC was produced on a Tandy 1000 IBM-compatible computer, initially using Volkswriter 3, a fine word processor from Lifetree Software, and mostly using the IBM/clone version of LeScript, my favorite word processor, from Anitek Software of Melbourne, FL. Previous issues were produced on the TRS-80, using the TRS-80 version of LeScript.

Reminder: We greatly appreciate your letters, solutions, questions, programs, improvements, contributions, subscriptions, problems, and gold bars. (Gold bars?? Just kidding to see if you're reading this far!) But remember: if you'd like a reply or acknowledgment, please include a self-addressed, stamped envelope. Thanks! P.S. Are your title, name, address on label all correct?

Bingo Solution/Program

In issue #2 we considered the question of the probability of an N-game in Bingo, meaning we wanted a simulation of bingo so that we see what fraction of the time we could expect a possible bingo win to occur by the N column before any other way of winning. See that issue for more details; a program follows.

Bingo Program for Estimating Frequency of N Game (TRS-80 Model 100 Portable)

```
10 CLS: PRINT "BINGO SIMULATION"
20 PRINT "FOR RECREATIONAL COMPUTING"
30 PRINT "DR. MICHAEL ECKER"
40 PRINT: PRINT "THIS PROGRAM WILL BE"
50 PRINT "A MONTE CARLO SIMULATION OF"
60 PRINT "AN N GAME IN BINGO (SEE TEXT)
70 INPUT "CONTINUE";C$
80 CLS: PRINT "COMPUTER WILL NOW PICK"
90 PRINT "NUMBERS AND LOOK FOR BINGO."
100 DIM C(16): REM 16 NUMBERS MAX
101 DIM P(5): REM B,I,N,G,O
102 DEFINT R
103 PRINT "GIVE ME 3-DIGIT # TO SEED THE RANDOM"
104 PRINT "NUMBER GENERATOR":INPUT RR
105 FOR A=1 TO RR
106 X=RND(100)
107 NEXT
108 CLS
110 R=INT(75*RND(1)+1): REM PICK NUMBER
115 IF K=0 THEN 150
120 FOR J=1 TO K
130 IF R=C(J) THEN 110: REM NUMBER NEW?
140 NEXT
150 K=K+1: C(K)=R
160 REM CLEVER WAY TO CATEGORIZE NUMBER
170 PO=INT((C(K)+14)/15)
180 PRINT MID$("BINGO",PO,1); C(K)
190 P(PO)=P(PO)+1
200 IF P(3)=4 THEN NGAME=NGAME+1: PRINT "AN N GAME": GOTO 250: REM N GAME?
210 IF P(1)=5 OR P(2)=5  OR  P(4)=5 OR  P(5)=5 THEN DOWN=DOWN+1: GOTO 250:  REM
FIVE UP/DOWN OTHER THAN N LINE?
220 PR=P(1)*P(2)*P(4)*P(5)
230 IF PR>0 THEN ACROSS=ACROSS+1:          GOTO 250: REM 1 OF B,I,G,O?
240 GOTO 110: REM CALL NEXT NUMBER
250 REM SET UP NEXT GAME
260 FOR Q=1 TO 5: P(Q)=0: NEXT: K=0
270 FOR J=1 TO K: C(J)=0: NEXT
280 PLAYED=PLAYED+1
281 PRINT "BINGO!"
282 PRINT "PERCENTAGE BREAKDOWN SO FAR"
283 NP=NGAME/PLAYED: DP=DOWN/PLAYED: AP=    ACROSS/PLAYED
284 PRINT "NGAMES: "; CSNG(NP) : REM CSNG CUTS VARIABLE DOWN TO SINGLE PRECISION.
285 PRINT "DOWN   : "; CSNG(DP) : REM YOU MAY USE NP, DP, AP IN THESE 3 LINES
286 PRINT "ACROSS: "; CSNG(AP) : REM IF YOUR MACHINE DOESN'T SUPPORT CSNG(variable).
288 FOR DL=1 TO 1000: NEXT: PRINT "NEXT GAME": FOR DL=1 TO 1000: NEXT
290 CLS:GOTO 110
```

Translation Notes in this box:
(Do not enter as part of program!)

CLS clears screen. You may need HOME or Print CLR or some such.

RND(100) produces a pseudorandom number less than 100. You may need something different for your machine - ditto with line 110's means of producing a random whole number from 1 to 75 inclusive. Consult your manual or ask a computer friend.

Line 180's MID$ command may have to be replaced by a SEG (?) command if you have a TI computer.

TI users may require extended Basic to enter multiple commands on a single line, and they may need a delimiter other than the colon for separating the different statements on a single line.

Entry of additional spaces on later lines is not necessary, as in 230 & 283.

Brief Explanation of Program: The program really begins around line 110 with the selection of a first ball (number 1 - 75 inclusive). It is tested if it is not the first ball to see whether the computer picked that number (i.e., that ball) already. If old, a new number is drawn; otherwise, it is entered and stored. Now the number is categorized in line 170 as to B,I,N,G,O column, the number is announced (as in "I22" or "N45") by printing on your screen, and the tally updated as to type of number, such as another N. Tests for possible theoretical bingo are made in lines 200 - 230, with N games announced/pointed out. If no bingo, the next number is called at line 240 by returning to line 110 (where the process continues with picking and testing number); if a bingo, the next game is set up in line 250. The computer announces the results thus far. Hit the Break key when you get bored.

A few readers sent in solutions, and the majority got values in the vicinity of around 12% or a smidge higher. The program shown I wrote on a TRS-80 Model 100 portable a while back. Equally good ones were contributed by many readers, and the only reason I am not publishing theirs is that I would have to key them in from scratch. The solvers include Ray McClanahan of New Orleans, LA; John Howell of Littlerock, CA, who would have offered to send a solution to readers if they needed it; and Maurice Servranckx of Canada. Each of these offered some personal insights, too. Readers: Keep those solutions and programs coming in!

Contributors

Many thanks for the financial contributions from these subscribers: Dr. H. Bruce Phillips, Mr. Maurice Servranckx (again!), and Dr. Frank J. Rosato, who also offered to spread the news about REC. Thank you all so much for giving because you believe in what we're doing, and without being asked!

Locate the Bomb!

A good reader sent in a nice, interesting idea. Although there is no signature, I believe the letter with the concept comes from Helmut Fuchs. (Please contact me, writer, if I am mistaken!)

Imagine a building with one million rooms, the building in the shape of a cube 100 x 100 x 100 (note that this does indeed give 1,000,000 cubic units). A bomb has been hidden in one of the one million rooms. You are equipped with a sonar-like device which will allow you to take a room and, from it, get a reading of the distance in units from the bomb. Each location is described by three coordinates (x,y,z), with each of x,y,z being between 1 and 100 inclusive. You need only imagine that there is an axis in each of the three directions for this last development.

With as few readings as possible (20? more? fewer?), the object is to locate the bomb so that it may be defused or removed.

As a two-issue challenge to REC readers, I'd like you to:

1) Write a program in Basic to allow you to play the game in which you take guesses, get feedback in the form of distances, and the computer keeps track of the total number of guesses you need before you locate the bomb (via its coordinates). Send me your best, shortest, simplest program which uses as few machine-specific features (such as peeks, pokes, calls, assembly-language subroutines, etc.) for possible publication. If you have a machine which uses TRSDOS or MSDOS you might send me a disk as well. Sample screendumps would be helpful for all other machines so that I may see what your program does.

2) Next issue, I'll publish at least one solution, and then the challenge will be a mathematical / recreational one instead, namely, how to find the bomb in as few guesses as possible. There are some preliminary indications that a good strategy might require about 20 guesses, but don't treat this as a hard and fast result. On the contrary, we'll be working together - all of us - on this one!

I look forward to writing my own program and reading all your programs as well. All contributors will receive acknowledgment in the next issue of REC, whether your program is used or not. (Space considerations don't allow more than one to be published; otherwise, we'd have to cut out other material.)

Paying for Postage

Last issue, I gave a not-so-hard problem which led to a Diophantine equation, which is to say, an equation whose solutions are restricted to integers and which has more variables than equations. In fact, a few readers said that the problem was too easy. (They did not seem to mind, but they just wanted me to know!)

Maybe so, but it is understood here in REC that if a problem is hard, then you do whatever you can to solve it, whereas if a problem is easy, you search for ways to find a solution which is more elegant. Elegance might include such features as shorter code, more readable code, more structured code. It is hard to define, although many of us recognize elegance when we see and admire it.

Therefore, although I was pleased with the many fine responses, I was a bit surprised that a few more of you did not use the opportunity to make a faster, better, or more elegant program than you did.

The problem again? You spend exactly $1 buying some combination of current US stamps, limited, at most, to the denominations of .14, .17, .22, and .39. Most readers recognized that this leads to 14a + 17b + 22c + 39d = 100, where a, b, c, and d represent, respectively, the number of 14, 17, 22, and 39-cent stamps.

Solvers included Helmut Fuchs (I think, as solution was with the "bomb idea" letter); John M. Howell (who also played with a variation of the original problem); Tom Neal (who also sent two other programs for other problems); Dr. Frank J. Rosato (who accidentally used 32 instead of 39); Dr. H. Bruce Phillips; Dr. Henry Sonneborn III; Maurice Servranckx; Steven L. Wentworth; and Loren Krienke. All solvers agreed that there are four solutions (a,b,c,d), namely (0,0,1,2), (0,1,2,1), (0,2,3,0), and (4,0,2,0). (The notation "(4,0,2,0)" means that a=4, b=0, c=2, d=0, with a,b,c,d as described in preceding paragraph.) With the exception of Mr. Krienke's, all the solutions, in essence, looked something like this:

```
10 FOR A=0 TO 7
20 FOR B=0 TO 5
30 FOR C=0 TO 4
40 FOR D=0 TO 2
50 IF 14*A + 17*B+ 22*C + 39*D = 100 THEN PRINT A,B,C,D
60 NEXT D
70 NEXT C
80 NEXT B
90 NEXT A
```

Note the nested loop structure. Of course, solvers usually had some embellishment to make the output more descriptive in my line 50, to clear the screen first, and so on, but at core, this is the common thread. Note how the solvers argue that the 100-cent ($1) total limits the possibilities to at most seven of the 14-cent stamps, for instance. However, only Loren Krienke of San Diego, CA, seems to have recognized an efficient modification. Mind you, in a simple program such as this with few solutions and small numbers, it may not make much difference, but it could with larger values. Interestingly, I was consulted about two years ago on a problem more involved than this but which had certain similarities, and the only way to make the program feasible, since there were 15 nested loops involved (!), was to use several devices, one of which is found in Loren's solution. In fact, however, there is also a second trick which Loren did not see, either. Let me describe both, for they are really not hard.

Suppose that your program, in the case of the postage problem, is already up to testing to see whether there might be, say, two 17-cent stamps. Then you must realize that you are even more limited in the number of possible 22- and 39-cent stamps, since you've potentially used up some of your money. Therefore, your loop limits can't really go as high.

Second, once you try a certain combination of other stamps, there is at most one possible number for the number of the last stamp needed to use up exactly $1. For instance, if the program is up to one 39, no 22's, and two 17's, with 14's worked on last (unlike the first program's order), then you've used up 73 cents, leaving 27 cents. You need only divide 14 into 27 to see the possible number of 14-cent stamps.

Third, to make this most efficient, make this last number the number of 14-cent stamps, not 39-cent stamps, for then you minimize the number of tests and loops.

```
10 FOR D=0 TO 100/39
20 FOR C=0 TO (100-39*D)/22
30 FOR B=0 TO (100-39*D-22*C)/17
40   A=INT((100-39*D-22*C-17*B)/14)
50 IF 14*A + 17*B+ 22*C + 39*D = 100 THEN PRINT A,B,C,D
60 NEXT B
70 NEXT C
80 NEXT D
```

In the new line 10, I used 100/39 instead of 2 just to set up the pattern. Basic will automatically use 2 since the loop has a default step of 1, meaning that D goes up by 1, going no higher than the value following the word "TO" in the "FOR ... TO ..." statement. Whatever value D has, you now have 39*D less available to spend, so you deduct it from your original 100 cents. Then, in line 20, you divide 22 in to see the maximum number of 22-cent stamps possible after the deduction. And so on with later lines until line 40, where A has just one possible value.

One last thing that I might mention is an alternative to the lines 40 and 50:

```
40  A=(100-39*D-22*C-17*B)/14
50 IF A=INT(A) THEN PRINT A,B,C,D.
```

Do you see why this works just as well, even if less obvious?

So, there were three or four things which could be done to make a better program, and only one solver did any of them (and then only one). As I mentioned, it might not matter much in a smaller program, but I'm sure other readers can back me up that it can make a big difference elsewhere.

Postage II

Just for fun, here is another postage problem. For this one, I would like solutions no longer than a half-dozen lines. There must be a good dozen different approaches, and this problem is one for which I will probably publish lots of solutions in an upcoming issue (if not the next).

First-class postage in the US costs 22 cents for the first ounce, and 17 cents for each additional ounce or fraction thereof. Thus, a 2-ounce letter costs 39 cents to mail, and a 2.001-ounce letter costs 56 cents. Write at least one program which allows user to input the weight in pounds and ounces (remember, 1 lb = 16 oz) and which then outputs the postage cost.

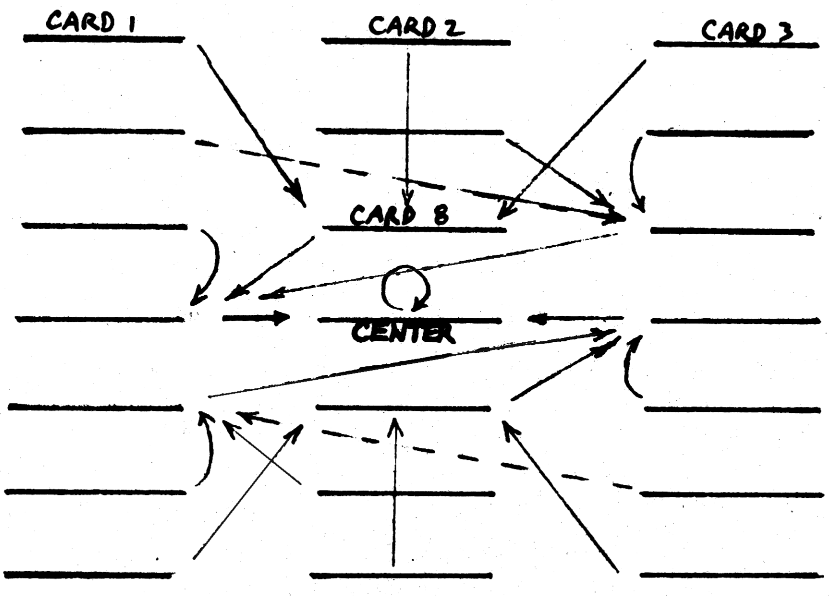
Strange Attractors and Black Holes (Part 3)

This month we look closer at the card trick in the previous issue. You will see how the middlemost card in the configuration operates in the manner of a black hole, as I defined it previously. See issue #2 for details on the card trick and other ideas.

Before I do so, let me acknowledge the computer programs for a computer version of the trick which were sent to me by Jorge Cadenas of Mexico, John Howell, and Maurice Servranckx. Mr. Howell reports doing the same trick sixty years ago, by the way (from which you may safely infer that those of you in your sixties and seventies are not alone with this wonderful computer obsession).

A couple of years ago, I did write out a mathematical proof for the trick. It involved writing a formula for the new position of a card picked after answering each question. That formula, by the way, involved the greatest integer function (the same INT function in your Basic), and it took some work. Here is a picturesque way of visualizing what happens, my second solution. A reader (Maurice Servranckx, I believe), sent in something similar in spirit.

What does the diagram mean? Each arrow indicates what happens if the card selected is the one at the end of the arrow, with the arrowhead indicating where in the layout that same card will appear next time. Don't forget that the selected card's column is given, so that a card in the upper left, for instance, can easily be deduced to wind up as the 8th card, as you sandwich the column identified between the other two columns. Think about the solution, and observe the role of the center, row 4, column 2, as black hole. In particular, note that it has both features: a card in that position winds up there after re-laying out cards, and other cards, if chosen, ultimately wind up there within a maximum of three turns of sandwiching and re-laying out of the cards.



THE 21 CARDS. WITHIN 3 MOVES (FOLLOW ARROWS), YOU'LL ALWAYS HIT THE CENTER.

Coming Next Issue: Program for "Locate the Bomb" - plus strategy of the game; Magic Squares and Magic Stars (courtesy of reader Maurice Servranckx, partially based on work of reader Dr. Henry Sonneborn III); postage programs and others from reader solutions; at least one more card trick (computerized or not); Software Reviews, including Templates of Doom, Night Mission Pinball, and at least one bridge program; and more of the usual goodies, challenges, recreations, programs, problems and surprises.

Software Reviews
Blue Lion Software's "Ticket to Paris"
(For IBMs and compatibles and Apples)

At PC Expo this July just past, I came across an interesting new software offering called Ticket to Paris, by Blue Lion Software. I was attending as a writer for Computer Shopper (see my account in the Sept. '86 Computer Shopper, albeit slighted botched by CS editorial). However, as my name badge showed, I was really officially there with free press pass as the editor and publisher of REC. Interestingly, just about the only product I found there of a less business nature was Ticket to Paris.

In fact, Ticket to Paris, in certain respects, epitomizes the kind of software this newsletter is named for: recreations and education. The version of Ticket to Paris which I reviewed was the PC version on the Tandy 1000. It features superb color and graphics on my color monitor, and I presume this to come from the actual filming the company did while in France to capture the appropriate images and scenes.

But what is the program? Briefly, it is a program which helps you with your learning of French - by your using it - while you simultaneously play an interactive game in which you seek to locate a cousin who left for Paris some months ago. In order to find him so as to persuade him to return home to the US, you must familiarize yourself with French life and culture, beginning with the moment you arrive at the airport. You may opt to use English, but if you choose French (my recommendation, if you know any), you are provided with a helpful dictionary. Caution though; the dictionary's pages tatter if you overuse it, although you may choose to use some of your limited resources to buy another.

In the course of all this, you have a fixed amount of time and a given amount of money in which to accomplish your mission. You can, however, save "games" in progress so as to allow you to continue at later sessions without starting over. The program allows about a half-dozen players. Of course, if you wish, you may start over with a new name.

The program presents numerous scenes of great realism in which you need to respond with some reply or possibly even action. You reply with the use of arrow keys to select from preprogrammed choices of responses. Such choices occur while you are in such places as hotels, museums, banks, restaurants, etc. There are icons and status indicators to give you feedback about where you are, how much money you have, how hungry you are, and so on.

To win the game, you need a certain number of points, and you must have followed up on a certain number of clues (the specifics are not important to mention here). You must do this within budget and time, without ignoring your bodily needs, and you must return to the airport - after paying your hotel bill, of course!

I've only played for a short while, but I found the program enjoyable and educational. It might not have that pressure-pounding, attention-riveting demand of arcade games, but Ticket to Paris is unique and has a nice feel to it.

The program seems to be copy-protected (I did not try a backup yet, but I noted that there is no mention of this topic, which frequently suggests that a program has been protected). Although I do understand the wish for such protection, I think that such should be stated explicitly if this is the case.

Nevertheless, I am pleased to offer a moderately high recommendation to Ticket to Paris, available for IBMs (including 256K jr and PC workalikes) and Apples; a Commodore 64/128 version is being prepared, too. Price (for any version) is $39.95, and product may be ordered directly from Blue Lion Software/ P.O. Box 650/ Belmont, Massachussetts 02178. (617) 489-2477. Note: a similar product, Ticket to London, has been announced by Blue Lion, and we hope to get a review copy of it, too.

COLLEGIATE MICROCOMPUTER - JOURNAL AND CALL FOR PAPERS

COLLEGIATE MICROCOMPUTER is a quarterly journal for the exchange of ideas on the role of the microcomputer in all subjects and areas of college and university life. Material appearing includes uses of hardware and software, descriptions of courses, units and topics using microcomputers, results of research using microcomputers, analysis of experiments using microcomputers, student projects, suggestions and tips, writeups of experiences as microcomputer consultants, reviews of the literature, evaluation of microcomputer use in the office and in materials preparation for teaching and research. Subscription rates are $28.00 per year and $36.00 per year for non-US subscriptions. For further information write: COLLEGIATE MICROCOMPUTER. Rose-Hulman Institute of Technology, Terre Haute IN 47803 USA.

PC Software Recommendation: Many readers are familiar with the concepts of commercial software, public domain software, and also the in-between category of shareware, in which you pay a nominal disk-copying charge for the full program (which is a copyrighted program still); if you like the program, you may register with the author or publisher and get such support as a manual, updates, and so on.

One of the people working in this area in the PC arena is Mr. Robert Nicolai, who writes interesting recreational and educational software involving poker and poker odds, word games, and many other nice programs which have received favorable reviews. Bob has extended a get-acquainted offer to REC readers with PCs or clones. I don't have the details because (sorry Bob!) I lost them when I moved my offices around, but for $5 plus $1 shipping, you get a nice demo of some fine programs. Write Robert Nicolai/ Cheapware/ 4038 N. Ninth St./ St. Louis, MO 63147, or give him a call at (314) 621-7618. Software producers and authors such as Bob deserve our support!

Well readers, it looks like another issue chock full of goodies, and it brings us to the point where 2/3 of most of your subscriptions have been filled. Many of you have responded warmly and most favorably to REC, with many pointing out the uniqueness of this newsletter and our mission - a mission which the big commercial publications will have little to do with, since there is no real money to be made.

But I need to hear from more of you. Do you like what you see and get here? How about the level? Would you like to see some more word-oriented challenges and recreations? (Yes, some are planned in the not-too-distant future!) Is REC something worth continuing for you and for me?

I have dispensed with the Marketplace section (except for the reciprocal ads or mentions on the previous page) just so that I could take a moment, without making hysterical claims the way many ad agencies do, to ask you to evaluate REC and decide to continue to participate in the fun, challenges, stimulation, and all the other features. (Heck, I didn't even mention that there will be more Strange Attractors and Black Holes coming up in a future issue! And moreover, the idea for it came from not just me but also from a reader!)

Although it is a bit early for some, I ask you to vote that you wish to continue by sending in your renewal now. Even though REC is one of the least expensive newsletters on the market, and has little or no advertising (more like no advertising), readers tell us that they cannot find anything like this anywhere else at <u>any</u> price.

So, I ask you to renew your subscription <u>now</u>. We're going to hold the line at $18.95 because we wish to thank you for your support and loyalty this past year. We don't ask for anything else, even though we're not making any real money on REC. If you care to contribute beyond that, we are most appreciative, but we do not pester you with renewal notices and appeals.

And don't forget, if you have not yet entered the first REC Sweepstakes, you may do so with your renewal. Just send your $18.95 check with your name, address, and computer model (needed for the awarding of sweeps prizes) on separate 3x5 paper. Drawing will be held Feb. 2, 1987.

Lastly, so as not to have to cut up your newsletter, I am not including a little order form at the bottom. You people are among the sharpest of consumers and are quite capable of placing such an order without the little form. Just remember to send to the address at the top of the first page.

Good luck in the sweeps to all, and I hope to hear from you all soon. Until next issue, may all your computing be both recreational and educational!

Mike