Recreational & Educational Computing Newsletter

Mar/Apr 1986 - Volume 1, Number 2.

Editor and Publisher: Dr. Michael W. Ecker. Executive Assistant: Susan R. Ecker.
Published six times per year at 129 Carol Drive/ Clarks Summit, PA 18411.
Subscription cost: $18.95 per year. Full reproduction rights $25 extra per year.

Editorial

Okay, okay! REC has been coming out consistently in the *first* week or so of the month immediately following the two listed on top. Why is this?

The fact is that we got started late with the first issue, and we've been merely "on schedule" - meaning once every two months - since then. However, with the summer coming and lots of other changes (e.g., no teaching), we plan to move up schedule back to where it really was supposed to be, namely in the middle of the two months listed.

Until then, let's pretend that we're creating a new publishing tradition: While every other publication known uses the following month's date, REC looks back at the last two months!

In other matters, we've also decided that we will indeed be looking at more "newsletter-looking" formats with other word processors. Although we never said it would be this issue, I was kind of hoping it would be. But, we will be looking into variety - if for no other reason, just to keep things lively.

Turning to finances, we have just about surpassed our first hundred in the number of REC subscribers. In an era where thousands and tens of thousands of subscribers are the norm, bear in mind that we accomplished this with virtually no advertising, but just direct mail to inquirers from magazine releases, and direct mail to past readers. Still, some people wonder how we manage. The answer is that we are dedicated to this as a labor of love as much as a money-maker. In fact, it is highly debatable as to whether we're really "making money" here.

Not that we're complaining. On the contrary, you readers are generous without even being asked. For instance, without any solicitation, Mr. Maurice Servranckx of Quebec sent a monetary contribution in addition to material for publication. He expressed the hope that the extra funds would help keep us going. We greatly appreciate it, but we want you all to know that, deep pockets or not, we have no intentions of leaving you in the lurch the way the big computer mags do when they become "insufficiently profitable"!

We do, however, ask you to keep your subscriptions current (but don't worry, we won't start nagging you to re-subscribe for 1987 for at least another two months). And please do keep sending solutions, questions, problems, etc. If you send disks, note: at the moment, we have only MSDOS and TRSDOS-based computers.

So, we're continuing happily to fill a niche in the "unprofitable" but computationally rich world of recreational and educational computing.

While publication dates printed on top are almost arbitrary, and the format can be explored and varied, the main thing is that you're getting your subscription. And, REC comes every two months, as promised, with the kind of number crunching, programming challenges, and other such reasons for which we presume you subscribed. Does it really matter what the date says on top?

Enjoy!

Mike

Michael W. Ecker, PhD
Editor/Publisher of REC

Note: When you write us - and we hope more of you do - if you would like a reply or acknowledgment, please enclose a self-addressed, stamped envelope. Thank you!

Solution: The Challenge of Self-Reference

Last issue I asked for a program which, when run, produces an exact listing of itself. We did not get a big response, but frankly, this is not an easy question. Recall, for instance, all the restrictions: No use of the List command, no use of I/O (that's input/output), no data files, etc. One reader sent a solution which used the saving of a file. No fair!

As promised, here is the solution of reader Norlin Rober of Marshalltown, Iowa. It is the most concise and elegant to date. I've edited it only slightly to allow it to run on most computers.

Here are details for the benefit of novices (and we have a few, including a certain noted and iconoclastic high school mathematics textbook author). Turn your computer on. Load the operating system, then load Basic. From Basic Ready, type in the program, entering a carriage return after each line.

Then type List (followed by a return). Look at the results carefully. Now type Run (followed by a return). Compare the two results. Isn't it remarkable?!

```
1 A$="1 A$=2 PRINT LEFT$(A$,5);CHR$(34);A$:PRINT RIGHT$(A$,51)
2 PRINT LEFT$(A$,5);CHR$(34);A$:PRINT RIGHT$(A$,51)
```

Analysis: Line 1 defines a string variable - a variable holding verbal / text data instead of numeric data. That variable is A$, the "$" indicating string variable. The rest of that line is the text data. You will note that most of it is the same as line 2 in line 2's entirety. The program then prints out portions so that the mission is accomplished - brilliantly! The LEFT$ and RIGHT$ functions pick out the left and right portions of the string A$ to the number of characters indicated (5 and 51, respectively, in this case). The "CHR$(34)" is the standard ASCII character code for the quote mark; it is needed to simulate the quote mark in running.

Congratulations, Norlin, on an elegant solution to a hard problem!

Editorial Update

This issue of REC is actually coming out two to three weeks later than originally planned. Rather than change the editorial, which had already been processed and made ready for printing, I've decided just to add this explanation.

Simply put, my TRS-80 Model 4P, on which I word process REC using LeScript from Anitek Software, had a problem. My local technician told me I'd have it back in a few days.

Well, a few days soon turned into a week. Parts had to be ordered. Soon it was two weeks.

Finally, after three weeks, I got the computer back (yesterday, as I type this). All seems to be fine now. I am sure you don't care for excuses any more than I do. The fact is, we've had an unusually high share of both good fortune and grave misfortune this past year or two. Some recent misfortunes have made it difficult to do much of anything, never mind the load involved. However, I'm not complaining, and I apologize for the lack of professionalism. I also won't subject you to any details. You paid for your subscription and you are entitled to getting a stimulating publication - preferably on schedule!

For my part, however, I prefer not to rush junk out to you just to make money.

Please stay with us and allow for our being human. If there's one good thing about adversity, it teaches sensitivity to the feelings of others. I am most concerned that you not feel upset with us.

Finally, I am pleased to report that our subscriber base is indeed working on its second hundred. True, this is little compared to the big mags. But bear in mind that the big mags cater to business interests. Remember, too, that we do little advertising, although we have received, and continue to receive mentions in other newsletters, math publications, computer publications, and elsewhere. Examples include Dr. Eydie Sloane's interesting <u>The Sloane Report</u>, about which I'll tell you more later, and the keep-you-up-to-date-on-business <u>PC Productivity Digest</u>, edited by Mr. Lawrence Oakley. I'll say more about PCPD later, but I'll confess that I have an extra connection to that publication: I was temporary editor of it last summer!

<u>Reminder</u>: Have a question? Is your mailing label 100% correct? Want a reply to something? Then please don't forget to include a self-addressed, stamped envelope. We communicate with all readers, but your cooperation is needed (if you need a personal reply). Please also remember that we actively solicit your programs, problems, challenges, comments, solutions, improvements, etc.

One Solution to Baseball Card Simulation

Last issue, I asked: Suppose that you were trying to collect a full collection of 50 distinct objects, such as baseball cards. If the cards are randomly distributed, you would probably get quite a few "extras" of several cards before you had at least one of each of the 50. About how many cards, on average, would you have to collect, to have a full collection?

As you can see on the other page, a solution is - as I suggested - to write a program to simulate the collection process and keep track of how many cards are needed. Since this varies per trial, do many trials - say 100 or more - and average the results.

The program shown is a mildly modified version of the first one received, sent to us by speedy programmer Merton L. Davis of Camden, SC. If memory serves, Merton is a chemist, but I think he's more of an alchemist, for he managed to squeeze in his solution on the back of a 14-cent postcard! (Some people will do anything to save 8 cents!)

Incidentally, I personally found this solution to be less intuitive than what seemed the natural way to go, but I felt that the first solver ought to get some recognition! I also received at least one disk, but I can't find it at the moment. If you sent me one, would you please remind me of who you are and what kind of disk so that I can take a look and also give you proper credit? The same applies if I fail to mention your name and you sent a solution on paper.

Other solvers - most with terrific solutions - included: computer science student John C. Shuey; Benjamin B. King, who also enclosed some interesting generalizations and other ideas; Helmut E. Fuchs, who also enclosed lots of graphs and other information about lots of other matters; John M. Howell (although most solvers got a higher value than he did); Ray McClanahan, who was right in there with his answer; Steven L. Wentworth, who contributed a disk as well as printout (thanks Steve!); Edward Bjorklund, who also contributed a schedule-generator program (tournaments?); loyal Canadian reader Maurice Servranckx; and H. Bruce Phillips, about whom, please read on.

I personally found the values on my TRS-80 Model 100 portable to be averaging between 225 and 230. Some readers found lower values on average; others found higher. Many computers were used. (Although I have over a half dozen, and am planning to acquire a Laser 128 Apple IIc clone, I used just the one.) So, who is right?

A mathematical solution is called for, but I did not have one. Thanks to Dr. Phillips, an engineer, I believe, the question has been settled: about 225, in theory. Here is his argument, and I quote:

"The first buy is a sure thing. I now need 49 more, and my chances of getting one is 49/50, so I'd expect to buy 50/49 cards on average. I now need 48 more, and I would expect to have buy 50/48 cards to get one of them..."

He goes on to say that this leads to 50*(1 + 1/2 + 1/3 + 1/4 + ... + 1/50) as the theoretical answer. I would just add that the argument is intuitive, yet quite sophisticated mathematically, so don't berate yourself if you have difficulty with this.

For the value of this, I entered this short program in Basic:

```
10 For K=1 to 50
20 S = S + 1/K
30 Next K
40 Print 50*S
```

and you should get, upon running, a value close to 225. (Line 20 keeps the running sum in variable S, as S is set equal to its previous contents, plus 1/K, where K is the present number of 1 to 50. The For-Next loop does the rest.)

Slightly modified version of first baseball / collection simulation program.

There were many variants - all very good - but almost all solvers got 225 as being in the ballpark of their final answer. (Sorry about the unwitting pun!)

```
1 R=VAL(RIGHT$(TIME$,2)): REM RANDOMIZE
2 FOR I=1 TO R*R:A=RND(1):NEXT
10 DEFINT B-Z:DIM B(50):T=1
20 N=1:B(N)=50*RND(1)+1
30 FOR X=2 TO 50:B(X)=50*RND(1)+1
40 FOR Y=1 TO X-1
50 IF B(Y)=B(X) THEN X=X-1:Y=X
60 NEXT Y
70 N=N+1
80 NEXT X:AS!=AS!+N:AV!=AS!/T
90 PRINT T;N;AV!
100 IF T=100 THEN END ELSE T=T+1:GOTO 20
```

Translation Notes:

Line 1 uses the time clock of the TRS-80 Model 100 portable for randomization. It takes the last two digits of the time and uses them in line 2 as a kind of synthetic seeding of the pseudo-random number generator. Without such a pair of lines (or something comparable), some computers - such as the Model 100 - will always produce the same results in every simulation, a clear contradiction to the notion of simulation with random elements!

If your computer produces a syntax error with lines 1 and 2, edit appropriately to suit your machine, or, if you don't know how to, simply omit them. (If you've already typed them in, from your Basic Ready, type 1 and touch the Enter key, and then type 2 and touch the Enter key. This will delete the two lines.)

Line 10 sets up variables with initial letters B through Z as integers, an option for speed. You'll need it since the programmer did not use an integer conversion in lines 20 and 30. The dimension statement in 10 is essential for the array whose values will be held in B(1), B(2), ... up to B(50). Line 20 places a random number in B(1), which is converted to integer by the aformentioned integer declaration. You may have to convert the 50*RND(1) + 1 to another form, such as RND(50) + 1, as on some TRS-80s, for instance. Similar remarks apply to line 30.

Lines 40, 50, 60 basically proceed to test the last generated random number to see whether it is new. Regardless, the counter N in line 70 is incremented by 1 (it keeps track of the number of cards or objects needed).

Single-precision variable AS! (that's what the exclamation point means) gives the total number of cards for all trials up to that moment, each trial being what is needed to get a complete collection. Note that, rather than averaging the individual results for each trial, this program takes the grand total number of cards and divides by the number of complete collections (or trials) to give AV!, the average. In fact, however, these are the same thing, so there is no problem.

In line 100, the second statement (the GOTO 20) is executed only if T<100, due to the ELSE. If this is confusing, leave off the colon and GOTO 20, make line 110 GOTO 20 - and all should be well.

Comments on Next Number Program in Last Issue

On page 4 of REC #1, I provided a program for a next number (not "the next number") in a sequence. Some comments are in order.

First, the Dec. 1985 Creative Computing column of mine, on which this was based, had some minor, inadvertent errors of exposition, but none of significance.

Second, I should mention when this program "works" - that is, in the "expected way". It does so when your formula matches that of a polynomial. Odd numbers fit this form, as we then have a function of form $f(n) = 2n - 1$, a first-degree polynomial (degree = 1). Perfect squares do as well, with form $f(n) = n^2$, a second-degree polynomial (degree = 2). One caveat, however, is that in order to get answers in the program which agree with the polynomial function (or formula) you have in mind, you need to specify enough of the first numbers of the sequence. This means that for a polynomial of degree N, you need to specify a minimum of N+1 numbers. If you fail to do so, there is a good chance that you will not generate the results you'd otherwise expect. (How does the program know what you really meant?)

So, if you think of the perfect cubes: 1, 8, 27, 64, ..., you need to specify at least four values (1, 8, 27, 64) in order to successfully generate the next perfect cube. You may offer more than four, if you wish: You'll still get the next cube after the last one offered.

Bingo!

You're in a crowded bingo hall. The numbers are slowly called out: B12... N33... G54... N41... B3... I27... N35... G48... N38.

"Bingo!" is the loud cry from a corner of the room. All eyes turn to the winner. Sure enough, the winner has won with five in a row using four Ns (numbers in the N column) and the free space.

"Damn! Another N Game!" complains the chain-smoker opposite me, while her companion nods knowingly.

While the above is not an exact verbatim transcript of any particular day's events, it happened enough in bingo halls way back in my more carefree days as a doctoral student to make me wonder: What is the probability of an N game?

By this I mean roughly: what is the probability that the eventual winner of a game will win by having four Ns plus the free space, as opposed to any other way? As stated, the question is imprecise, so let me sharpen it. Imagine that you are in a bingo hall so crowded that every possible card is in at least one person's hand. Hence, at the first possible moment when somebody could theoretically win, we imagine that at least one person does win.

Math-minded people: How many cards are possible? Consider cards different if they fail to be absolutely identical in every respect. This is not a new problem, but it is a nice one if you've never tried it before.

If memory serves, the number of possible different bingo cards is over 5 E 26 (5 followed by 26 zeros)! In practice, far fewer cards are printed, perhaps just a scant several thousand, but for various reasons, the results might not be seriously affected by having only a few hundred players with just these few thousand cards.

The possible ways to win - that is, get five in a row vertically, horizontally or diagonally - are to get, minimally : five of any one of B, I, G, O (vertical) or one of each letter B, I, G, O and the free space (horizontal or diagonal), or four Ns and the free space (the other vertical line possibility, our N game). Of course, it does not matter if you have other numbers called as well. It is assumed that whichever comes first determines the classification of the game. I also trust that you readers are already familiar with this game.

In 1975, in a mathematical problem proposal published in the Pi Mu Epsilon Journal, I asked for the mathematical probability of an N game. For many years, the problem went unsolved, not because it was difficult, but rather because it was tedious. Within the past year or two, however, a spate of reader solutions appeared in that publication. In the intervening time, however, it occurred to me that this makes a nice programming challenge for a computer simulation.

So, here is the challenge. Write a program to simulate the random selection from the 75 bingo balls. Assume that as soon as a game could possibly end, it does. Then see whether or not it was an N game. (You might want to see what other kinds of game you might wish to classify.) Repeat the simulation several hundred or so times, and have the program keep track. The fraction of the time that an N game results should indicate its likelihood. Then you can see whether or not the woman had due cause to groan. (As with most gamblers, most bingo players lose much more than win, so groaning is a natural state for them).

Query: How many of you readers prefer to have some hints? How many of you would like to see an introduction to the rudiments of programming? How many think it would be a waste of space? Please let me know your preference. (No vote means it does not matter to you that much, and I have your okay to do as other readers say, and as I think.)

Strange Attractors and "Black Holes" (Part 1)

Long-time readers of "Recreational Computing" in Popular Computing probably recall the interesting problem of narcissistic numbers. That problem was to find all positive whole numbers which equal the sum of the cubes of their digits. For instance, 153 has this property, as 1x1x1 = 1, 5x5x5 = 125, 3x3x3 = 27, and their sum is 1 + 125 + 27 = 153, the original number.

Actually, there is narcissism of various orders, with the above third-order narcissism. If you have never solved this before by computer, you might like to take that as a programming challenge right now. One hint I'll give is not to worry about numbers over a few thousand. (For theoretical reasons I won't get into this issue, there is none that large.)

The problem can be generalized and even extended to card tricks, which I'll do in a future issue (probably next issue). For now, let me put the matter into perspective with another concrete example, this one due to Dr. Jack Lamb, a good, long-time reader of "Recreational Computing" (in both Popular and Creative) and now REC. Perhaps in mimickry, mathematician Lamb asked for all numbers with the property that the original number equals the sum of the factorials of the digits.

Let's break that down a bit: The factorial of a natural number n (or positive whole number n) is n! (read: "n factorial") and defined as n(n-1)(n-2)...(3)(2), so that 3! = 3x2 = 6, 4! = 4x3x2 = 24, etc. Also, 1! = 1, and by special convention, 0! = 1.

There are many similarities to a solution. Dr. Lamb produced a mathematical bound on how high to try to look (about which I'll try to dig up the information for next issue), and then looked at all numbers up to the upper bound. To make things quicker, he considered shortcuts as to whether some kinds of numbers could not be "sum-factorial numbers" - numbers equal to the sum of the factorials of the digits.

Can you produce at least one sum-factorial number?

Remark: The methodology is very similar to that for narcissistic numbers.

After dealing with this problem, I went on to look at the two problems here (narcissistic numbers, sum-factorial numbers) in another way. What if you start with another number and it's not one of these special numbers? What if you apply the process to the answer? For instance, using 27, if you take the sum of the cubes of the digits, you get 351, not the original 27. If you now take the sum of the cubes of the digits of 351, you get 153, not 351. But if you now do this process to 153, as we saw above, you do get 153 back again.

So, I asked Jack (when I met him in person in New Orleans at a math convention this past January, which seems an eternity ago!) to consider the question I'm asking for his sum-factorial numbers as well. In fact, I ask readers to consider playing around with both types of questions from this dual perspective of what I call "black hole attraction". The two elements are as follows:

A black hole (with respect to some process) is a number which:
1) produces itself again upon application of the process, and
2) tends to be such that repeated application of the process will eventually draw other numbers to it.

In the narcissism example, 153 is a kind of black hole, as it produces itself, and if you take any integer multiple of 3, and apply the process repeatedly, you eventually must hit 153 (at which point, you keep getting 153). The number 153, by the way, is sometimes called Kaprekar's Constant, and a program involving this appears on the TRS-80 version of Magic Math Plus. See ad for Magic Math Plus under "Marketplace" section; if enough MSDOS users are interested in the program and Magic Math Plus, I'll add it to the collection, upon request, at no extra charge.

Almost forgot: If the original number is not a multiple of 3, then you will not reach 153. Many possibilities then abound...

Okay, readers: Observations? Solutions? Programs? Comments? Extensions and generalizations? More variants? Send 'em in! More next issue...

Letters, Communications, Comments, Miscellany

One or two readers offered to send in something about rounding, precision, accuracy, etc. Please do! Also, Helmut Fuchs sent me a small but nice list of suggested math recreation materials for Richard Barth and others interested. Send SASE to me for a copy.

A non-subscriber offered to send in a program in Pascal for getting the day of the week from the date. Such a program in Basic appeared a year and a half ago in an article I wrote as a (then) contributing editor for Soft Sector magazine, a Sanyo-specific publication, which is how the reader came to write me. It also appears, revised, in the MSDOS version of Magic Math Plus (see "Marketplace"), but several questions: 1) Should REC include non-Basic programs? 2) Should it include this topic? 3) Should it include anything not written by a subscriber?

Follow-up quip: Mr. Benjamin B. King - also known as B.B. King - tickled my funny bone after my noting his name in REC #1. Said Mr. King in a letter to me: "I once gave my name to a Holiday Inn reservation clerk who said she knew how I felt.... Her name was Carol Burnett."

Coming Next Issue!

The Mystique of Strange Attractors and Black Holes, Part 2. More Challenges and Programming, Including Goodies from Readers. Solutions to Previous Questions. Letters and Help. Software Reviews. Other Newsletters of Interest. (We did not have room in this issue, so we'll give the information next one.) More.

Quick Bytes: Software and/or Book Reviews

(In this section, REC Newsletter looks at some meritorious programs in the recreational or educational areas, for various computers, on a rotating basis.)

Let's Play Monopoly!

Custom Software's TRS-80 Model 100 Portable adaptation of the classic game

In a few words, LPM is a beautiful implementation of Monopoly for the Model 100. It comes on cassette tape, and requires 24K or more memory.

On the screen's left, a small version of the board is drawn. On the right, space is available for identification of properties and similar information. In the middle of the board, dice stand poised, ready for "rolling" via a random number generator. Seven options avail themselves: roll dice, buy, mortgage, unmortgage, trade, review property holdings, & quit.

I was surprised to find my heart thumping as I began. "It's just a stupid game on a little computer!" I reminded myself. Why was I getting so excited? Was it the realism? The sound generation accompanying the movement of the tokens? Was it finally having an opponent who liked to move as quickly and unhesitatingly as I did?

I was absolutely delighted and astonished to find that I could save my game in progress. I had not realized this until the time to quit!

If you enjoy Monopoly and have a Model 100, you'll love Let's Play Monopoly. You won't love the memory/storage problem involved with all the RAM LPM takes up, unless you invest in a disk drive. But you'll love the carefully thought-out features, including the Inkey-type routines for easy responses, terrific and careful error-trapping, the nice graphics, the options for buying, mortgaging, unmortgaging and the like, and the fun. You'll also love the fact that saving a game no longer means keeping a board with money and properties somewhere where somebody can disrupt it or cheat, as the computer takes care of all. It keeps track accurately, too, and handles all transactions such as rent payments, fines, and windfalls automatically such as by Chance and Community Chest.

I would personally think that a few dollars less might be a more appropriate price, but an enormous amount of work seems to have gone into Let's Play Monopoly. Custom Software deserves the reward for such an excellent product.

Let's Play Monopoly, on cassette tape for the TRS-80 Model 100 portable with 24K (32K recommended), not copy-protected, $29.95 plus $1 shipping/handling, to

Custom Software/ 1308 Western/ Wellington, KS 67152. Phone (316) 326-6197.

Marketplace

All orders or requests for information to: Recreational Mathemagical Software/ 129 Carol Drive/ Clarks Summit, PA 18411. Please specify computer model.

Bridge-86 for MSDOS and TRSDOS (including tape) computers features computer play of opponents, bidding, and one-keystroke responses, plus you always get the best hand. The program allows replaying, displaying of cards, printing of hands, computer scoring, and more. Best features-to-price ratio around at $18.95 cost.

Magic Math Plus is Recreational Mathemagical Software's main product of recreations and "mathemagic" and is available for $5 off to REC subscribers. It includes number tricks, computer investigations, computer math, computer games, and much more. The TRS-80 version lists for $37.50 and has nearly 40 programs on a self-booting disk in menu-generated format. The Sanyo and MSDOS versions come with nearly 20 programs for $27.50 list, both with menu-generated format. The Sanyo 550/555 version comes on a self-booting disk. Titles include Base 2 Trick, SuperBlackjack, the Game of N, and much more. At the risk of even more immodesty than usual, let me quote Carl Springett (letter to me, May 24, 1986): "Dear Dr. Mike... Thank you for sending Magic Math Plus so quickly..." After some other business, he concluded: "For my money, Magic Math Plus is worth twice what you are getting for it!" Thank you, Carl. I had a co-author for TRS version, and this made our day! (Mind you, this is for the TRS-80 version at the full $37.50 price! Letters and other unsolicited testimonials on file; photocopies will be sent to any reader who encloses a SASE.) $22.50 or $32.50 to REC subscribers.

Gin! Hollywood, Oklahoma, knock, beautiful color, help. This neat little program has it all (IBM-compatible only). Now just $28 postpaid. (Reduced from $37.45)

Math Tutor I. If you have an IBM, MSDOS, or Sanyo computer, and want a tutorial for arithmetic, this is one of the best I've seen. $39.95 postpaid, with unusually strong documentation. Specify generic MSDOS or Sanyo 55x series.

Annuity financial planning for retirement, for MSDOS and TRSDOS computers. $20.
Fastloan II loan amortization, also for MSDOS and TRSDOS computers, $18.
Financial program combo on tape for TRS-80 Model 100 portable, just $9.

RMS is now a distributor for software, various computer and other books on making money (usually computer-related), directories such as the zip code directory, and mailing lists of computer magazine editors. Inquire, please.

REC Order Form

To: Dr. Michael Ecker, Editor of REC.
Dear Mike: Yes, please keep REC coming. My $18.95 is enclosed.

Title (optional), Name ________________________________
Address ____

$18.95 per year. Make checks payable to Recreational & Educational Computing Newsletter/ 129 Carol Drive/ Clarks Summit, PA 18411. Phone: (717) 586-2784.
For an extra $25 per year, schools may purchase unlimited reproduction rights.
To all subscribers: Have A Healthy and Happy Time! Until next time.... Mike....