EDMONTON 99'er COMPUTER USERS' SOCIETY

P/O Box 11983"
Edmonton, Alberta
Canada T5J 3L1

99'er ON LINE... is the news letter of the Edmonton 99'er Computer User's Society published ten times a year. Unless otherwise stated, all articles may be republished in other news letters provided that source and author are identified. We will credit authors quoted in 99'er ON LINE.

CORRESPONDENCE: Newsletter editor: Bob Pass, 59 Labelle Cr, St. Albert, Alberta, Canada T8N-2G6. (403)458-7658. All other correspondence should be sent to the address at left.

OFFICERS: president Tom Hall, vice-pres Ken Godbeer, treasurer Jim Mulligan, secretary Roxanne Appelt

DISCLAIMER: Information published in this newsletter is created by and for amateurs; therefore, we cannot guarantee the accuracy or use of presented information.

REGULAR MEETINGS... of the Edmonton 99'er Computer User's Society are held on the second Tuesday of each month in room 849 of the General Services building of the University of Alberta from 7:00 till 10:00 PM and are open to all members in good standing. Non-members may attend their first meeting free of charge.

ADVERTIZING... Commercial space is available in this news letter at the following rates: full page $20.00, half page $15.00, 1/4 page $10.00. Discuss your needs with Jim Mulligan at 467-6021, at the next meeting, or send "photo ready" copy to the P/O Box above. Members may advertise their personal computer related items for free but are asked to limit their ads to about 50 words. Mail your ads to the editor's address or hand it to him at the general meeting; newsletter deadline 3'rd Monday of the month.

MEMBERSHIP FEES: Family 12 months, $20.00, 6 months, $15.00. Students 12 months, $15.00, 6 months, $10.00. New member initiation, $20.00.

OCTOBER MEETING

Minutes of September executive meeting were adopted as read.

The next general meeting falls on Remembrance Day, Nov.11'th. There was some uncertainty regarding access to the building. Subsequent enquiries indicated that the building will be open.

Ken Godbeer reported that the following are additions to the disk library. Note that some are "freeware" which will require user donation to the author.

FASTERM & PRBASE -- both version 2.
SIDEPRINT -- new version.
FUNNEL WRITER -- version 3.3

Subscriptions were taken for CLUBLINE/99. The order was sent off Thursday, October 16'th. With luck, the first issue should arrive for the December meeting.

The Ottawa Users' Group sent us a copy of the video tape of their TI fair held last spring. The first part, a speech and demonstration of Myarc's Geneve, was shown.

Jim Mulligan gave an introductory demo of FASTERM and how to access our bulletin board. Jim handed out a "fact sheet" on FASTERM which is appears to be quite a bit better than the documentation included with the FASTERM disk.

NEXT MEETING

The next meeting will be Tuesday, November 11'th at 7:15 PM. Same place as usual; General Services Building, U of A campus in room 849.

Tom Hall will continue the demonstration on accessing our bulletin board with emphasis on up and down loading. Also, we will show the next segment of the Ottawa TI Fair video.

FEATURE PROGRAM

Located elsewhere in this month's newsletter is a program listing of a game written by Jim Beck, a young member of our group. The following is Jim's documentation.

DESERT WARFARE

By Jim Beck

Note: Extended BASIC and joysticks required.

As with most of my programs, this one is, you guessed it, a game. The object of the game is to defend your trench from enemies attacking from all sides. There are many different enemies for you to blast. There are enemy foot soldiers which run at you, tanks that roll down the road toward you, and helicopters that come in at an angle. If an enemy soldier gets close to the smaller trench on the side of the screen, he may decide to jump into it. When he does, it frees both his hands for throwing grenades so beware!! You may shoot the soldiers, tanks, helicopters, and grenades to keep them from reaching your trench. If they do, they kill you and the game ends. As the game progresses, the enemies get faster and harder to hit.

To control your soldier simply push up or down on the joystick to move and push left or right to aim in different directions. To shoot, push the fire button. Your gun will continue firing until you let go of it.

NEWSLETTER DISKETTE AWARDS

This newsletter is almost entirely made up of submissions by our members. Articals by Michal Jaegermann and Bob Burley and a program from Jim Beck. For their efforts, they receive a free diskette and the eternal thanks of your editor.

If you too would like a diskette, simply pick up a newsletter disk from me at a meeting, place your artical, program, or what have you on it and return to me. Upon publication, you will receive your disk.

Items you may consider are your favorite program (make sure it is yours or freeware), articals you found interesting in other newsletters, magazines or newspapers, grapevine news, product reviews, etc.

EDMONTON AREA BULLETIN BOARDS

By Bob Burley

Located elsewhere in this newsletter is a list of Bulletin Board Systems that I have compiled. They are mostly from listings on other systems, but also from just about everywhere else that I ran across a BBS number. Any systems I called and found that they were no longer operating have been deleted from the list, however there just isn't enough time to verify them all. Every week a BBS somewhere ceases operation, but it seems like two more start up at the same time. For this reason you must pay attention to the hours of operation if listed, although the best policy is to call an unfamiliar system for the first time at a decent hour.

If you should have any information regarding additions, deletions, or corrections pertaining to this list, please let me know at the regular club meeting, or write a short letter to the newsletter, or even leave a message for me on the TI-BBS (operated for the club by Tom Hall).

THE JOY OF DRAWING RECURSIVELY

by Michal Jaegermann

A few months I showed you how to use Forth to get the dragon curve. You will find here two more examples. For references and more explanations - please refer to the previously mentioned book "The Fractal Geometry of Nature" by Benoit B. Mandelbrot. Both designs for this month's, as well as the dragon, are recursive. Do not think that all fractals are of this nature. Far from it. Recursion is simply a tool which allows a creation of rather intricate pictures without writing a lot of code. So Bob will not kick me out of the newsletter and you will have enough patience to type in programs. Moreover, with a little bit of imagination it is possible to extend these images far beyond the confines of the resolution of your display.

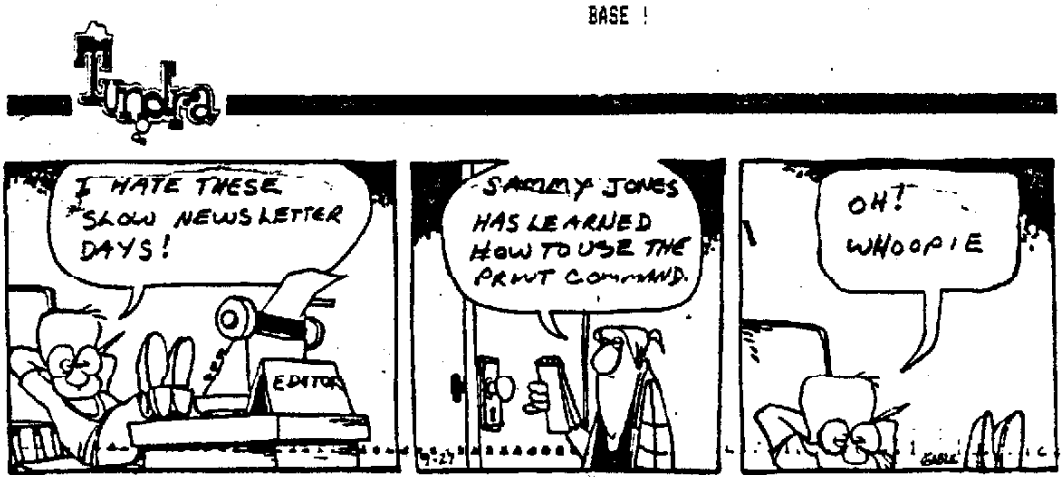
I think that I talked long enough in the previous article. Lets go to the crux of the matter. The first program (left, below) is the code which draws a 3/2 dimensional coastline of a quadratic Koch island - whatever the meaning of that may be. It assumes that you have loaded the first screen, which defines SEGx's, of the dragon curve code - presented in my previous artical on Fractals. As a matter of fact definitions of variables and SEGMENT are also the same as before - so you may skip them if they are already loaded.

The other curve (right hand listing) is a little bit different. It is created by taking out pieces of a surface. What is left after an infinite process of this kind is the Sierpinki carpet. It consists mainly of holes, but it still has enough substance in it to be in one piece. The only problem is that we cannot wait infinitely long until the weaving process is finished. So I settled for some approximations.

Have you other neat ideas for fractal drawing? If so, then how about submitting the next fractal article to continue this series.

```
( Quadric Island / Mandelbrot "FGN", p. 50 ± MJ 6JUN86 )
( load SEG's from DRAGON       -GRAPH -GRAPH2 -TEXT )
BASE @ HEX ODDHD VARIABLE SEGTBLE
0 VARIABLE HEADING  0 VARIABLE XCOR  0 VARIABLE YCOR
: SEGMENT ( x y -- x1 y1 )
    HEADING @ 06 AND SEGTBLE @ + @ EXECUTE  ;

: SHORE_LINE  ( level -- )
    -DUP IF 1-  \ drop level by one
            DUP MYSELF  2 HEADING +! DUP MYSELF -2 HEADING +!
            DUP MYSELF -2 HEADING +! DUP MYSELF
            DUP MYSELF  2 HEADING +! DUP MYSELF  2 HEADING +!
            DUP MYSELF -2 HEADING +!     MYSELF
         ELSE XCOR @ YCOR @ SEGMENT YCOR ! XCOR ! ENDIF ;

-->
( Quadric Island / Mandelbrot "FGN", p. 50 ± MJ 6JUN86 )

: QSETUP ODDHD SEGTBLE ! 2 HEADING ! 80 XCOR ! 98 YCOR !
    10 DCOLOR ! GRAPHICS2 07 7 VWTR DRAW ;

: Q_ISLAND ( level -- )  3 AND 80 OVER DUP + 1+ SRL STEP !
    4 0 DO DUP SHORE_LINE -2 HEADING +! LOOP DROP
    KEY DROP ;

: ISLANDS  QSETUP 4 0 DO I Q_ISLAND LOOP TEXT  ;

BASE !
```

```
( Sierpinski carpet ± 1st scr ± Michal Jaegermann 5JUL86)
0 CLOAD CARPETS BASE @ DECIMAL -GRAPH -GRAPH2 -TEXT
3 VARIABLE EPS
: BOX  ( x y size -- ) SWAP 2DUP + SWAP
    DO  2DUP OVER + SWAP DO I J DOT LOOP LOOP  DROP DROP ;
: (CARPET) ( x y size -- )  EPS @ OVER >
   IF  BOX
   ELSE  3 / >R
     2DUP R MYSELF
     OVER R + OVER R MYSELF
     OVER R DUP + + OVER R MYSELF R +
     2DUP R MYSELF OVER R DUP + + OVER R MYSELF R +
     2DUP R MYSELF
     OVER R + OVER R MYSELF
     SWAP R DUP + + SWAP R> MYSELF
   ENDIF ;          -->
( Sierpinski Carpet 2nd scn ± Michal Jaegermann  5JUL86 )

: CARPET ( eps -- )  EPS !
   GRAPHICS2 7 7 VWTR 47 0 162 (CARPET) ;

: CARPETS     486 EPS !  16 DCOLOR ! DRAW
   BEGIN EPS @ DUP 2 > WHILE 3 / CARPET
         20000 0 DO LOOP
         PAUSE IF TEXT ;S ENDIF
   REPEAT  DROP KEY DROP TEXT ;

BASE !
```



```
10 CALL CLEAR
11 CALL COLOR(12,2,2)
12 CALL HCHAR(1,1,120,32)::
CALL HCHAR(24,1,120,32):: CA
LL VCHAR(1,1,120,24):: CALL
VCHAR(1,32,120,24)
20 CALL SCREEN(11)
30 DISPLAY AT(10,8):"DESERT
WARFARE"
40 DISPLAY AT(12,9):"  FREEW
ARE  "
50 DISPLAY AT(14,9):"BY JIM
 BECK"
60 DISPLAY AT(21,9):"PRESS A
NY KEY"
70 CALL KEY(0,K,S):: IF S=0
THEN 70
100 RANDOMIZE
110 SCR=0
120 TIC=0
130 CALL COLOR(8,2,2):: CALL
 COLOR(9,2,2):: CALL COLOR(1
2,2,2)
140 CALL CLEAR
150 ENS=3
160 NKJ=1
170 CALL SCREEN(2)
180 CALL COLOR(1,2,2)
190 CALL MAGNIFY(3)
200 CALL CHAR(40,"303078F8FC
7F7070705048488600000000000
0000000000000000000000000")
201 CALL CHAR(100,"000001000
1010303030100000000000000000
0008080C0C0C080000000000000"
)
202 CALL CHAR(104,"001F1F0F0
809CBFE0B09080F1F1F000000FFF
FFF01F1F979F9F101FFFFF0000"
)
210 CALL CHAR(130,"804020100
8040201010204081020408"")
220 CALL CHAR(132,"000402010
0000000102040000000000629
49EF0F0F0F0F8646250602070207020"
)
230 FOR DELAY=1 TO 30
240 CALL HCHAR(1+INT(24*RND)
,1+INT(32*RND),96,6)
250 NEXT DELAY
251 FOR DELAY=1 TO 15
252 CALL HCHAR(1+INT(24*RND)
,1+INT(32*RND),120)
253 CALL HCHAR(1+INT(24*RND)
,1+INT(32*RND),108,2)
254 NEXT DELAY
260 CALL CHAR(92,"0C0C1E1F3F
FE0E0E0E0A12121100000000000
0000000000000000000000000000")
270 CALL SPRITE(#1,40,2,96,1
28)
280 CALL CHAR(120,"3E4181818
181916E")
281 CALL CHAR(108,"000480020
09000010")
290 CALL CHAR(124,"000804020
10000000001030202020200080C
06C3C387070E0C0808080808000"
)
300 CALL COLOR(12,2,16)
310 CALL COLOR(13,8,11)
320 CALL VCHAR(2,15,120,22)
330 CALL VCHAR(2,18,120,22)
331 CALL HCHAR(13,19,108,14)
:: CALL HCHAR(14,19,108,14):
: CALL HCHAR(15,19,108,14)
332 CALL HCHAR(12,19,120,14)
:: CALL HCHAR(16,19,120,14)
340 CALL HCHAR(1,16,120,2)
350 CALL HCHAR(24,16,120,2)
360 CALL VCHAR(2,16,130,22)
361 CALL VCHAR(8,4,120,4):: 
CALL HCHAR(7,3,120):: CALL H
CHAR(12,3,120)
362 CALL VCHAR(8,3,131,4)
370 CALL VCHAR(2,17,131,22)
380 CALL CHAR(96,"00C8255ABC
3C1818")
390 CALL COLOR(9,3,11)
400 CALL HCHAR(15,8,96):: CA
LL HCHAR(6,3,96):: CALL HCHA
R(20,23,96):: CALL HCHAR(7,3
0,96)
410 CALL SCREEN(11):: CALL C
OLOR(1,2,11)
420 BR=20+INT(150*RND)
430 TOP=1+INT(5*RND):: IF TO
P=2 THEN CALL SPRITE(#2,132,
13,1,60,-30,4):: GOTO 450
431 TOR=1+INT(8*RND):: IF TO
R=5 THEN CALL SPRITE(#2,104,
2,100,256,0,-16):: TOP=2 ::
GOTO 450
432 UIO=1+INT(4*RND):: IF UI
O=2 THEN CALL SPRITE(#2,40,2
,8*8,256,0,8):: GOTO 450
440 IF BR>90 THEN CALL SPRIT
E(#2,40,2,BR,1,10-INT(20*RND
),ENS)ELSE CALL SPRITE(#2,92
,2,BR,1,10-INT(20*RND),-ENS)
450 ENS=ENS+1 :: IF ENS>30 T
HEN ENS=30
460 CALL JOYST(1,X,Y):: IF X
=4 THEN CALL PATTERN(#1,40):
: NKJ=1 ELSE IF X=-4 THEN CA
LL PATTERN(#1,92):: NKJ=-1
470 CALL MOTION(#1,-Y*8,0)
471 CALL DISTANCE(#2,8*8,4*8
,FST):: IF FST<180 THEN CALL
 DELSPRITE(#2):: CALL HCHAR(
8,2,40):: CALL HCHAR(8,3,42)
:: GOTO 9000
480 CALL POSITION(#2,JOP,POJ
):: IF POJ>116 THEN IF POJ<1
30 THEN IF JOP<175 THEN IF J
OP>16 THEN 660
490 CALL KEY(1,K,S):: IF K=1
8 THEN 510
500 GOTO 460
510 CALL SOUND(1,-5,0)
520 CALL POSITION(#2,BR,GKJ)
521 CALL SOUND(1,-7,0)
530 IF NKJ=1 THEN IF GKJ<128
 THEN 560
540 IF NKJ=-1 THEN IF GKJ>12
8 THEN 560
550 CALL POSITION(#1,G,H):: 
IF BR+16>G THEN IF BR-16<G T
HEN CALL SOUND(50,-7,0):: GO
TO 570
560 GOTO 500
570 IF TOP<>2 THEN CALL PATT
ERN(#2,124)ELSE 650
571 IF TOP=8 THEN CALL DELSP
RITE(#2)
580 CALL MOTION(#2,0,0,#1,0,
0)
581 CALL HCHAR(8,2,32,2):: C
ALL HCHAR(9,2,32,2)
590 FOR DELAY=1 TO 30 STEP 4
600 CALL SOUND(-99,140-DELAY
,0)
610 NEXT DELAY
620 SCR=SCR+250
630 CALL DELSPRITE(#2)
640 GOTO 420
650 CALL COLOR(#2,9):: CALL
SOUND(300,-6,0):: GOTO 580
660 CALL MOTION(#2,0,0,#1,0,
0):: CALL PATTERN(#1,124):: 
CALL SOUND(100,131,0):: CALL
 SOUND(50,131,0):: CALL SOUN
D(700,156,0)
670 IF SCR>HISCR THEN HISCR=
SCR :: CALL COLOR(8,2,1)ELSE
 CALL COLOR(8,2,1)
680 DISPLAY AT(10,8):"HISCOR
E=";HISCR
690 DISPLAY AT(12,9):"SCORE=
";SCR
700 DISPLAY AT(15,9):"PRESS
ANY KEY"
710 CALL KEY(0,K,S):: IF S=0
 THEN 710
720 GOTO 100
9000 CALL HCHAR(9,2,41):: CA
LL HCHAR(9,3,43):: CALL SPRI
TE(#2,100,2,9*8,2*8,4-INT(8*
RND),10):: TOP=8
9001 FOR DE=1 TO 15
9002 CALL SOUND(1,990+DE*9,0
)
9003 NEXT DE
9010 GOTO 430
```

EDMONTON BULLETIN BOARDS

Abyss, The 473-6889
?????.

Alnazar Bulletin Board 494-9655
C-64, Private, Pseudonyms, 24 Hrs.

Antithesystee 464-4172
Apple II, Pseudonyms, 24 Hrs.

Bird's Nest 469-7155
Apple, 24 Hrs.

Boss #1 484-5707
Multi-user, Charges fee 24 Hrs.

Boss #2, Conferencing 484-5716
Multi-user, Charges fee 24 Hrs.

Boss #3, 300 Baud 484-5720
Multi-user, Charges fee 24 Hrs.

Boss #4 484-2845
Multi-user, Charges fee 24 Hrs.

C.U.E.BBS (BOB's Commodore) 466-7656
Private, 9 PM.-6 AM./ 24 Hrs Wknds.

C128 PC Information System 470-0788
(Dungeon System I) ?????.

Camelot II 437-4793
Semi-private, 24 Hrs.

Coco-Base, The 439-4131
Coco, Public, 24 Hrs.

Comline 458-1435
?????.

Command Post 435-2762
?????.

Commodore Connection 461-3842
C-64, 8 Message Boards, 24 Hrs.

Comline 458-1435
?????.

Custom Micro Systems BBS 456-0974
Apple AE-Pro,6 PM-8 AM,24 Hrs Wknds

Dark Side BBS, The 458-2651
?????.

Datacca, (The Third Reich), 437-7389
Apple, 24 Hrs.

Datapac (300 Baud) 420-0185
24 Hrs.

Datapac (1200 Baud) 423-4463
24 Hrs.

Diamond Head 454-3618
?????.

Domain, The 454-1921
Apple, 1200 Baud, ?????.

Dragonworld I 462-2196
C-64, RPG, X-MODEM, 24 Hrs.

E.A.C.H., (Atari Club), 461-2428
Atari 400, Pseudonyms, 24 Hrs.

E.C.C.S. 435-5260
CPM, Public, 300/1200 Baud, 24 Hrs.

E.T. Board, The 487-0154
BBS Info, Pseudonyms, 10 PM.-10 AM.

Edmonton Commodore BBS 470-0796
C-64, 7:30 PM. - 9 AM.

Edmonton RCP/M 454-6093
CPM, X-MODEM, 24 Hrs.

Epson North Users Group BBS 486-3183
300/1200 Baud, 24 Hrs.

Eric's BBS 466-0155
C-128/C-64, 24 Hrs.

Evergreen BBS 986-2165
Tandy 1200, 300/1200 Baud, ?????.

First Reich 438-5734
Apple, X-Rated, 24 Hrs.

Friar Tuc's 464-3802
IBM, Sporadic Hrs.

IBM YAPCUS 986-4025
IBM, 6 PM. - 2 AM. / 24 Hrs Wknds.

Jim's C-64 456-8557
C-64, 24 Hrs.

KCBBS (Kelly's Computer) 483-1935
Apple, Private, ?????.

Kevin's BBS 489-8079
C-64, ?????.

Knight's Quarters, The 458-2290
Apple, Pseudonyms, 24 Hrs.

Koko 437-1722
?????.

London Drugs BBS 489-1277
?????.

MTS (University of Alberta) 432-4801
24 Hrs.

Mafia, The 435-9468
Atari, 300 Baud, 24 Hrs.

Marauder BBS, The 454-5598
Apple, Pseudonyms, ?????.

Martian Chronicles, The 464-2723
C-64, Pseudonyms, 12 Mdnt. - 4 PM.

Meadowlark CP/M plus RCP/M 435-6579
CP/M, 24 Hrs.

Michelle's BBS 467-4923
?????.

No-Name BBS 464-0511
Public, 5 PM. - 8 AM.

Northern Alberta Coco 474-0147
TRS-80 Coco, 24 Hrs.

Northern Alberta IBM PC 487-7019
(New Number 436-9130??1?? ?????.

Odd Space 986-9332
Pseudonyms, 24 Hrs.

PC Board 489-1603
IBM, 24 Hrs.

P.T.L. BBS 457-2203
Texas Instruments, Religous 24 Hrs.

Parkland County RCP/M 962-0328
300/1200 Baud, 24 Hrs.

Party Line, The 434-3567
Apple, Pseudonyms, 24 Hrs.

Paul's Stereo Board 467-7933
?????.

Pyramid 421-7014
Coco Club, ?????.

Rat's Nest 986-2961
?????.

Rock, The 986-8430
?????.

Satellite RBBS-PC 474-5262
300/1200 Baud, ?????.

Selectdisk Software BBS 489-0931
C-64, ?? 9 PM. - 9 AM. ?? 24 Hrs.??

Software Order Desk 425-0218
?????.

Solutions, Pascal Programs, 459-3877
Turbo Pascal,7 PM-7 AM,24 Hrs Wknds

Southside RCP/M 463-5774
IBM, CPM, 24 Hrs.

@Spires@AEline 436-4929
?????.

Starbase 12 462-2196
?????.

TI-BBS 424-3238
Texas Instruments, TI Club, 24 Hrs.

Valhalla 461-1874
Atari, 300/1200 Baud, 24 Hrs.

Warlords 484-5104
C-64, 10 PM. - 10 AM.

Warren's Commodore 435-2642
C-64, ?????.

West World North 471-2827
Apple, 24 Hrs.

Western Canada Dist. Centre 461-6960
CPM, $20/Yr., 24 Hrs.

Westminister Place BBS 435-0184
300 Baud, 9:30 PM. - 3:00 PM.

Wizard's Castle 435-1413
Atari, ATASCII only, 24 Hrs.

Workshops 420-1679
?????.

Moth first of many notorious computer bugs

NEW YORK (AP) — On Sept. 9, 1947, a moth flitted through an open window at Harvard University and made computer history.

Somehow, the moth became pinched between the points of an electrical relay in the Harvard Mark II computer and caused the primitive machine to spit out wrong answers.

A technician found the offender, removed it with a pair of tweezers and taped it into the computer's log book with a notation: "First actual case of bug being found."

U.S. Navy Rear Admiral Grace Hopper, who worked on the Mark II, says that moth was the world's first computer bug. From that day on, she says, any time the computer failed, operators would say they were "debugging" it.

And there have been some notorious bugs.

One memorable bug at the Vancouver Stock Exchange caused the computer to chop off tiny fractions from the stock index instead of rounding to the nearest number. Bit by bit, the index lost about half its potential value before the bug was discovered late in 1984.

And astronauts making the first moon landing in 1969 were given a scare when the software aboard Apollo 11 sent an alarm indicating it was overloaded. It turned out the instructions on how to operate the computer were faulty.

Another time, a misplaced comma in a NASA program sent a Voyager spacecraft toward Mars instead of Venus and the mission had to be aborted.

Bugs, though, were a term for mechanical failings even before computers existed.

Thomas Edison was bedeviled by bugs nearly a century ago, says a newspaper article of the time cited in the supplement to the Oxford English Dictionary.

"Mr. Edison, I was informed, had been up the two previous nights discovering 'a bug' in his phonograph — an expression for solving a difficulty, and implying that some imaginary insect has secreted itself inside and is causing all the trouble," said the article in the March 11, 1889, edition of the Pall Mall Gazette.

The Harvard Mark II was built at Harvard, then dismantled in 1948 and moved to the Naval Surface Weapons Centre in Dahlgren, Va., where it was used until 1956 for calculating bomb and missile trajectories.

A small museum at the base that displayed the log book with the bug taped in it was closed within the past few years to make room for office space. The book is in a secured building now, but the base's public-affairs office will consider special requests to see it.

And although big computer programs often contain more than 100,000 lines of code, unaided programmers are still able to write only about 20 lines a day, said Bill Daverne, a spokesman for Netron Inc. in Toronto, a company that offers to automate the process.

In fact, companies typically have to wait more than two years to get software from the time they ask their programmers for it, a survey by Computerworld magazine shows.

With computers growing so complicated, designers are being forced to treat bugs and other computer problems as facts of life. Many of the 90 computers that help fly a Boeing 767, for example, duplicate efforts so that if one produces a wrong answer it can be "outvoted" by others.

And the business community can take some of the blame.

Some developers, seeking a competitive edge, rush programs and hardware to market before they have spent the time and money to debug them thoroughly. They let customers catch the bugs, then release repaired versions.

"I can assure you that software development is a humbling experience," Patrick McGettigan, president of Landmark Systems Corp., said in a recent letter to Computerworld. "No matter how hard you try, no matter how skilled you are, you will fail in some fashion."