THE CORE WAR NEWSLETTER

The Official Newsletter of the International Core War Society

Features

Another Version of Core War *by Bernd Menzel* **2**

Letter From the Director *by Jon Newman* **5**

Articles

The 1991 ICWS Tournament Results *by Mark A. Durham* **8**

Electronic Update *by Mark A. Durham* **9**

Columns

The MADgic Corner **14**

Quarterly Challenge **15**

Volume 5, Number 3 Summer 1992 An ICWS Publication

Another Version of Core War

by Bernd Menzel

[*Editor's Note: Many people at various times have suggested modifying Redcode to more closely resemble "real" assembly languages with multiword instructions. Mr. Menzel has actually* implemented *this idea and submitted this article on September 11, 1990. I apologize for the delay in its publication.]*

Redcode is rather different from other assemblers. The code of one instruction is always stored in one memory word. Most of the instructions can change only the lower part of a memory word. In this article I want to introduce another version of Redcode which is closer to real assemblers.

It was in the middle of April 1990 when I heard about Core Wars for the first time. I was fascinated by the idea of two programs fighting against each other. So I decided to implement Core Wars on an IBM-compatible computer with an 80386 processor, and after some experiments I designed my own version.

In the original version the command code, address modes, and two addresses are coded in one large number. You can change either the whole register or the part representing argument B.

In my implementation each memory word consists of two bytes only. As a consequence, more than one word is necessary to represent a complete instruction. Therefore I put the command code and the address modes in one memory word and the arguments in the following words.

The command code is built up as follows: bits 8-15 represent the instruction code, bits 4-7 represent the number of words affected, bits 2-3 represent the address mode of the A-operand, and bits 0-1 represent the address mode of the B-operand.

The command **MOV 0, 1** for example is not coded in one address but in three:

address	old coding	my coding
512	MOV 0 1	MOV
513	??? x x	0
514	??? x x	1

Encountering a data instruction, Redcode now takes the argument of the command and inserts it into the code in the same way as every assembler does.

```
                        JMP
       JMP label         3
       DAT 22           22
label  ADD #1,-1        ADD
                         #1
                         -1
```

This, of course, causes certain other problems:

• Commands occupy different numbers of words (e.g. three words for a move instruction and one word for a data instruction).

• With the commands using more than one memory word you always have to increment the program counter by more than one (usually by three).

• The simple war program Imp (**MOV 0, 1**) does not run any longer because it moves its own instruction code to its first argument, then jumps to the next instruction which may be invalid.

```
→MOV   →MOV    MOV
  0      MOV    MOV
  1       1      1
 ???     ???   →???
```

Therefore a modification of the move command was necessary.

The syntax is now:

MOV*n *arg1, arg2*

where *n* is a natural number greater than zero representing the number of memory words to be moved at once. Thus the Imp is simply rewritten as the follows:

```
MOV 0, 1  ⇒  MOV*3 0, 3
```

Of course you can also extend other commands to such multiword instructions, e.g. the compare instruction CMP.

A closer look shows that the multiword instructions are more powerful than the original commands. Obviously they can do things at once where their older cousins need a loop.

To show this, let us compare an old version of the program Twin with a new version. Twin generates a copy of itself at a different memory location and then jumps to that location.

```
; Twin, original version
p1     DAT    0
p2     DAT    97
start  MOV   @p1,   @p2
       CMP    p1,   #9
       JMP    exit
       ADD   #1,     p1
       ADD   #1,     p2
       JMP    start
exit   MOV   #97,    91
       JMP    91
```

```
; Direct translation
p1     DAT    0
p2     DAT    97
start  MOV*3 @p1,   @p2
       CMP    p1,   #15
       JMP    92
       ADD   #3,     p1
       ADD   #3,     p2
       JMP    start
```

With the help of the powerful multiword instruction you can write an even shorter version of twin:

```
; Twin 2
start  MOV*5  0,     97
       JMP    94
```

In order to punish programs using these powerful commands, a `MOV*`*n* command will use *n* cycles with *n* ranging from 1 to 15.

The first strategy by which this may be achieved consists of reading the instruction one time and storing it in some kind of accumulator. During the following cycles it is checked if the instruction is completely done.

This solution has one advantage (or disadvantage, depending on the point of view). If a multimemory instruction is moved into the accumulator, the program is in a safe state for the corresponding number of cycles. During those cycles, its opponent is not able to destroy the executing code.

The other strategy is to split the program counter into two parts: cycle counter and memory pointer. The memory pointer represents the address of the instruction in memory. The cycle number denotes the number of cycles that already have been processed. The instruction is read and the cycle counter is incremented.

If the cycle counter equals the counter included in the instruction code, the statement is completely done. Then the cycle counter is reset to zero and the program counter is moved to the next instruction.

We have already talked about the programs Imp and Twin in this version of Redcode. Now I want to show further examples and compare behavior in battles to the original versions.

First, let us look at a program called Dwarf. It puts a zero at every tenth memory location and hopes to hit its victim this way.

```
ptr    DAT    -1
start  ADD   #10,    ptr
       MOV   #0,    @ptr
       JMP    start
```

Another example, the Imp Pit, uses the SPLit command. It will alternately move an Imp and put zeros before its start point to stop other Imps coming from behind.

```
ptr    DAT    0
loop   MOV   #0,     ptr
       JMP    loop
start  SPL    loop
       MOV*3  0,     label
label  DAT    0
```

Comparing some of these programs in 100 battles, we achieved the following percentages of wins:

	Original	New
Imp:Dwarf	20%:17%	50%:50%
Imp:ImpPit	0%:29%	8%:29%
Twin:Imp	42%:45%	33%:55%
Twin2:Imp	N/A	38%:49%
Twin:ImpPit	1%:35%	26%:64%
Twin2:ImpPit	N/A	27%:47%

Obviously, you can draw two conclusions from these numbers:

1. There are fewer battles without a winner.

2. Small programs now have a better chance to win a battle.

The reasons are simple: a program needs more memory because a single instruction uses more than one memory word. So the larger a program was in the original version, the more memory it occupies in the new version. On the other hand, larger programs will need fewer cycles to execute once:

	Memory		Cycles	
Program	old	new	old	new
Imp	1	3	1	3
Dwarf	4	9	3	3
Imp Pit	3	8	3	5
Twin	10	18	48	29
Twin2	N/A	5	N/A	5

Conclusion: Programs become larger and are more sensible so they are easier to destroy and there are fewer battles without a winner. The multiword instructions allow more efficient, faster versions of already existing programs without destroying the balance totally.

Letter From The Director

by Jon Newman

This has been a quiet quarter, which has given me time to sort out the mailing database and to try to contact potential new members. Our notices on major electronic bulletin boards have attracted some attention. In addition, I received a large packet of forwarded mail from Mark Clarkson, former Director of the ICWS. Letters continue to pour into Mark's mailbox due to a reference in A. K. Dewdney's book The Armchair Universe (*Editor's Note:* W.H. Freeman and Company, 41 Madison Avenue, New York, NY 10010. ISBN: 0-7167-1939-8). All of those people, from Finland to the Republic of South Africa, now have information on the ICWS.

It turns out that Mr. Buckley has incomplete data on which members received the ICWS'88 Standard. Therefore, every one of you should receive a copy of the current standard with this mailing.

The financial news from this first mailing cycle is good. Even after the US$200+ cost of that mailing, the ICWS is still running a slight surplus. The US$15 membership fees appear quite adequate for now.

There is bad news, however. Most of the listed memberships from William Buckley's database expired with the last mailing, and most of those people have not

renewed. This is understandable given the long delay since the last mailing, but it is sad to see the membership drop so sharply. I hope my readers here, who by and large joined more recently than the others, will be sticking with Core War and the ICWS.

Even worse news is that 23 issues of the Spring 1992 TCWN came back as undeliverable. If you have a more recent address for any of these people, please let me know right away (these are, of course, their old addresses):

Almergren, Joachim
4401 West Verdugo Ave, Ste. X
Burbank CA 91505

Balla, Aaron
1230 Barrington Street
Halifax, Nova Scotia
Canada

Colin, Lawrence S.
440 Richmond Park
Richmond Heights OH 44143

Ediger, Bruce A.
4 S. Ogden Street, No. 44
Denver CO 80209-2327

Elziere, Vianney
45 rue D'Ulm
75005 Paris, France

Feldman, Alexander
Rudolphstraβe 72
A-4040 Linz
Republik Osterreich

Hanβgen, Stefan
Kropsburgweg 6
D-7500 Karlsruhe 21
Bundesrepublik Deutschland

Jensen, Ole Hogh
Hellevangen 12
DK-9210 Aalborg 50
Danmark/Denmark

Kirchner, Alexander
Rebenring 61
D-3300 Braunschweig
Bundesrepublik Deutschland

Leonard, Paul N.
PO Box 3205
Redwood City CA 91710

Malpelet, Jean-Denis
59 rue de la Chine
75020 Paris, France

Ohnheiser, Laurence J., Jr.
12351 Marshall
Chino CA 91710

Paludan, Ronald V.
1117 West Enid Avenue
Mesa AZ 85210-3438

Pelly, Richard F.
6521 Lennox Drive
Huntington Beach CA 92647

Roberts, Mark G.
MSU Union POB 6852
East Lansing MI 48826

Rowland, Ernest N., Jr.
16901 Larkspur Lane, Apt. 5
Independence MO 64055

Sabadash, Gary
700 Irolo Street, No. 22
Los Angeles CA 90005-2330

van der List, Michael J.
669 Washington Street, No. 1
Brookline MA 01246

Wade, Fred
PO Box 716
Atwood CA 92601

Weil, Heinz Joachim
Ursrainer Ring 101
D-7400 Tubingen
Bundesrepublik Deutschland

Zupan, Marc
32616 Belle Road
Avon Lake OH 44012

Finally, a reminder: The 1992 Core War Tournament will be held in less than six months, so be ready!

Jon Newman

P.S. I recently received a letter asking about the disclaimer I sometimes place in mass mailings. I think my reply is important enough that it should be reprinted here:

Dear Sir:

When I said in my form letter that Microsoft might not approve of my ICWS activities, I meant this tongue-in-cheek and not seriously. I have no reason to believe that Microsoft would really disapprove of the ICWS. This sort of humorous disclaimer is common in Internet ".signature files", but clearly less common in the real world. I suppose I should be more careful about making this sort of joke in a mass mailing.

I am serious, however, when I say that Microsoft has nothing to do with the ICWS and carries no responsibility or liability for it. The ICWS is strictly my personal business rather than Microsoft's business.

I absolutely do not want anyone contacting e.g. Microsoft technical support or public relations with Core War questions, problems, or complaints. These people have better things to do. It would look bad for me if someone from phone support had to track me down to resolve this. Most importantly, Microsoft takes good care of me, and I try not to get it in trouble.

I hope this answers your question.

Sincerely,

Jon Newman
Director, ICWS

The 1991 ICWS Tournament Results

by Mark A. Durham

The 1991 ICWS Tournament consisted of warriors from both the Spanish Branch Section and the Electronic Branch Section.

The Spanish Branch Section ("Sociedad Española de Redcode") preliminary tournament, the "1er TORNEO de REDCODE", was carried out the end of the first week of December 1991 by J. J. Merelo of "Departmento de Electrónica y Tecnologia de Computadores, Universidad de Granada, Granada (Spain)". The tournament was well covered in *Micro Mania* magazine, both before and after, and received some support from Apple Computer. The poster for the tournament is a computer graphic feast for the eyes. I will try to print it (in color) in a future newsletter.

Winners in the Spanish preliminary were Cristobal Jiménez López and Juan A. Molina Corpas with first place Arania, Jose María Badía García's R.I.P. in second, and Juan Manuel Rodríguez Ibañez's KK was third.

The Electronic Branch Section's preliminary tournament was dominated by the two German Stefans with Stefan Röttger's XTC and Stefan Hänβgen's Quarter. There was also a strong showing by both entries of the team of David Bofinger and T. J. Allen, Sonic Kobold and Vlad (the Impaler).

Here are the 1991 ICWS Tournament results:

Program	Wins	Losses	Ties	Points
XTC	25	7	0	75
Fernandez	22	7	3	69
Vlad	18	10	4	58
Arania	17	9	6	57
Quarter	17	12	3	54
Laevsl	15	9	8	53
Electro	15	11	6	51
Sonic K.	15	16	1	46
KK	13	13	6	45
Ayla	11	13	8	41
R.I.P.	12	17	3	39
Minador	9	13	10	37
DAC	5	17	10	25
Influenza	3	17	12	21
Spider	3	17	12	21
Milikk	1	13	18	21

Some of the Spanish entries were altered slightly because of incompatibilities with the ICWS'88 standard. For instance, several warriors which used `SPL 0 B` (an ICWS'86 construct) were changed to `SPL @0 B` to run properly. In each case, the alterations were inspected to insure that the warriors would only benefit from and not be hindered by the changes.

Electronic Update

by Mark A. Durham

For those of you without access to electronic resources such as email, ftp, and news on Usenet and Internet, you may be feeling a little left out of the game. After all, you are missing out on the Core War archive and the "King of the Hill" server. And you are missing the daily discussions of Core War strategy and listings of Redcode programs available in the news group rec.games.corewar.

So how can you gain access to all of these goodies if you are not already well connected through your school or business? There are several ways. Possibly the least expensive way to gain easy access is to get an account on an open access Unix site. There are many of these and most are listed in a periodic posting to the alt.bbs newsgroup (Subject: Nixpub Posting). This does present a Chicken/Egg dilemma if you can not get someone else to get the list for you.

If you have the necessary computing "muscle" and the appropriate UUCP software, you may consider tying your machine into the networks as its very own node. You will need to get a "feed" from somewhere. Feeds may be available locally at no charge to you, or you may need to pay a company for the service.

Your local BBSes are also a possible method. Many are well connected with the rest of the world through various methods, including Fidonet. Check out several computer magazines (especially local ones) for lists of nearby BBSes.

More expensive are commercial computer services. Currently, the best connected of these is Portal. Portal offers an online service with worldwide email, Usenet news groups, local SIGs (special interest groups), and file archives for several platforms. CompuServe, GEnie, and the other online services offer varying amounts of more limited access. The services (and prices) of these companies frequently change because of competition.

~~~~~~~~~~~~~~~~~~~~~~~~~~~

From Antti J. Karttunen of Helsinki Finland (karttu@mits.mdata.fi) I received a message about "Core War" in Finnish. The literal translation is *ydinsota* from *ydin* (core) and *sota* (war). When you translate *ydinsota* back into English you get "nuclear war"!

~~~~~~~~~~~~~~~~~~~~~~~~~~~

For those of you with email but not ftp, there is an ftp server at

`ftpmail@decwrl.dec.com`

which will allow you to transfer

files through email. Send a message with a Subject: and body text consisting solely of “help” (without the quotation marks) for more information.

~~~~~~~~~~~~~~~~~~~~~~~~~~~

For those of you with email but not access to Usenet and the newsgroup rec.games.corewar, there is a Usenet mail server available. In order to receive rec.games.corewar through email, send a message with a Subject: and body text consisting solely of “subscribe” to

```
rec-games-corewar-request
  @ucbvax.berkeley.edu
```

To unsubscribe, simply change the message to “unsubscribe”. This method should work for any newsgroup. Just remember to substitute “-” for “.”.

You can post to rec.games.corewar by sending your message to

```
rec-games-corewar
@ucbvax.berkeley.edu
```

Again, this method should work for other newsgroups as well.

~~~~~~~~~~~~~~~~~~~~~~~~~~~

If you are interested in Tierra, you will be interested in the following paragraphs condensed from the Tierra Digest, Volume 1, #9:

The (Tierra) C source code creates a virtual computer and its operating system whose architecture has been designed in such a way that the executable machine codes are evolvable. This means that the machine code can be mutated (by flipping bits at random) or recombined (by swapping segments of code between algorithms), and the resulting code remains functional enough of the time for natural (or presumably artificial) selection to be able to improve the code over time.

Along with the C source code which generates the virtual computer, we provide several programs written in the assembler code of the virtual computer. One of these was written by a human and does nothing more than make copies of itself in the RAM of the virtual computer. The others evolved from the first, and are included to illustrate the power of natural selection.

The operating system of the virtual computer provides memory management and timesharing services. It also provides control for a variety of factors that affect the course of evolution: three kinds of mutation rates, disturbances, the allocation of CPU time to each creature, the size of the soup, etc. In addition, the operating system provides a very elaborate observational system that keeps a record of births and deaths,

sequences the code of every creature, and maintains a genebank of successful genomes.

The operating system also provides facilities for automating the ecological analysis, that is, for recording the kinds of interactions taking place between creatures.

This system results in the production of synthetic organisms based on a computer metaphor of organic life in which CPU time is the "energy" resource and memory is the "material" resource. Memory is organized into informational patterns that exploit CPU time for self-replication. Mutation generates new forms, and evolution proceeds by natural selection as different genotypes compete for CPU time and memory space.

Diverse ecological communities have emerged. These digital communities have been used to experimentally examine ecological and evolutionary processes: e.g., competitive exclusion and coexistence, host/parasite density dependent population regulation, the effect of parasites in enhancing community diversity, evolutionary arms race, punctuated equilibrium, and the role of chance and historical factors in evolution. This evolution in a bottle may prove to be a valuable tool for the study of evolution and ecology.

The Tierra V3.13 source code; and the source code and DOS executables of all tools are available now. Please note that the source code in the ftp site and the source code provided on disk will each compile and run on either DOS or UNIX platforms. It is exactly the same source code in either case.

The DOS executables are available only on disk, and can not be freely distributed.

The complete source code and documentation is available by anonymous ftp at:

tierra.slhs.udel.edu [128.175.41.34]
life.slhs.udel.edu [128.175.41.33]

in the directories: almond/, beagle/, doc/, and tierra/.

To get it, ftp to tierra or life, log in as user "anonymous" and give your email address (eg. tom@udel.edu) as a password. Be sure to transfer binaries in binary mode (it is safe to transfer everything in binary mode).

Each directory contains a compressed tar file (filename.tar.Z) and a SRC directory that contains all the files in raw ascii format. You can just pick up the .tar.Z files, and they will expand into the complete directory structure with the following commands:

```
uncompress tierra.tar.Z
tar oxvf tierra.tar
```

If you do not have ftp access you may obtain everything on DOS disks by making a check for $65 (US dollars drawn on a US bank) payable to Virtual Life. Specify 3.5" or 5.25" disks. Send the check to one of the following addresses:

Tom Ray (Jan. through Aug.)
Santa Fe Institute
1660 Old Pecos Trail
Suite A
Santa Fe, NM 87501

Virtual Life (Sept. through Dec.)
P.O. Box 625
Newark, Delaware 19715

The DOS disks contain everything but ALmond (ALmond can be provided on disk by request, but it only runs on a Unix platform). The disks include DOS executables, source code and documentation. The DOS disks include an easy installation program. This is the same source code available in the ftp site. If you have ftp access, there is no need to buy the disks.

There are two mailing lists. The first list is for people who only want to get the official announcements, updates and bug-fixes. The other will carry the official postings, and are intended for discussion of Tierra by users.

The lists are:

• tierra-announce: Official updates, patches and announcements only

• tierra-digest: Discussion, updates, etc. (digest form)

The addresses are:

tierra-request@life.slhs.udel.edu

for the list administrator (Tom Uffner). to be added, removed, or complain about problems with any of these lists,

tierra-digest@life.slhs.udel.edu

to post to the list, and

tierra-bug@life.slhs.udel.edu

for bug-reports or questions about the code or installation.

You may also be interested in the Artificial Life mailing list. Subscribe to the list by sending a message to:

alife-request@cognet.ucla.edu

Post to the list by sending a message to:

alife@cognet.ucla.edu

Tierra has been mentioned in many varied places, but there are several papers specifically about Tierra.

Ray, T. S. 1991. ‘‘Is it alive, or is it GA?’’ Proceedings of the 1991 International Conference on Genetic Algorithms, Eds. Belew, R. K., and L. B. Booker, San Mateo, CA: Morgan Kaufmann, 527-534.

Ray, T. S. 1991. ‘‘An approach to the synthesis of life.’’ Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, vol. XI, Eds. Farmer, J. D., C. Langton, S. Rasmussen, & C. Taylor, Redwood City, CA: Addison-Wesley, 371-408.

Ray, T. S. 1991. ‘‘Population dynamics of digital organisms.’’ Artificial Life II Video Proceedings, Ed. C.G. Langton, Redwood City, CA: Addison Wesley.

Ray, T. S. 1991. ‘‘Evolution and optimization of digital organisms.’’ Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers, Eds. Keith R. Billingsley, Ed Derohanes, Hilton Brown, III. Athens, GA, 30602, The Baldwin Press, The University of Georgia.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~

For those of you interested in C–Robots, you can write to the game author and copyright holder

Tom Poindexter
6864 Amherst Ct.
Highlands Ranch CO 80126
tpoind@uswest.com

C–Robots is a game in which you write programs, in the language of C, to control one of several robots in an arena. The object is to seek out, track, and destroy the other robots while preserving your robot from their attacks.

Your programs are aided by hardware functions for scanning, starting and stopping drive mechanisms, firing weapons, and so on. A battlefield display is included for watching battles (robots, missiles, explosions, etc. along with status reports) take place.

For those of you who prefer not to program in C, there are also spin-off versions of C–Robots in other languages such as Pascal.

Some of the archives where you will find C–Robots:

wuarchive.wustl.edu (128.252.135.4)
/systems/ibmpc/msdos/c/crobots.arc
/systems/amiga/fish/fish/f3/ff345/CRobots.zoo

alex.stacken.kth.se (130.237.237.3)
/disk1/pc/c/crobots.arc

athene.uni-paderborn.de (131.234.2.32)
/pcsoft/amiga/games/tactics/crobots.lzh
/news/amiga/bin/volume90/fun/crobots-2.3

nic.funet.fi (128.214.6.100)
/pub/msdos/languages/c/crobots.lzh

sabrina.dei.unipd.it (147.162.2.106)
/pub/ibmpc/games/crobots.zip

src.doc.ic.ac.uk (146.169.3.7)
/ibmpc/wsmr–simtel20.army.mil/c/crobots.arc
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The MADgic Corner

Welcome once again to my corner! It has been an interesting summer for myself, and I hope for you as well. In this issue's column, the usual mix of news and Redcode.

You should have received this issue while it is still technically Summer. Be that as it may, many of you will have gone back to traditional "Fall activities" such as school and the flavor of Summer has already left your palate.

This is just my long, drawn out way of saying that despite this issue arriving in a timely manner as promised, precisely three months after the last issue, that I want to shift publication back one month. Therefore, you should receive your next issue in the middle, not the end, of Fall.

This also means that if you have something to submit for publication that you should do so right away. I do not have any reader submissions currently waiting for publication. Without your submissions, the Fall issue could be very thin.

If you do wish to submit an article, letter, cartoon, or suggestion, you may do so by directly contacting me. I moved over the summer, and after the phone company changed my number twice and the postal service changed my zip code I think I can give you reliable information.

Submissions to:

The Core War Newsletter
c/o Mark A. Durham
18 Honeysuckle Terrace
Spartanburg SC 29307-3760
Phone: (803)579-0431

or via email to:

durham@cup.portal.com

~~~~~~~~~~~~~~~~~~~~~~~~~~~

The 1992 annual ICWS tournament is coming up sooner than you think. Here is the vital information you need to submit a warrior to that event (subject to revision in the Fall TCWN):

Deadline: December 15th, 1992.

Rules: ICWS '88, core size of 8192, maximum number of processes per warrior of 8000, and ties declared after 100000 cycles (turns per side).

Submission: Send entries to Jon Newman in one of these formats (in order of preference):

• Electronic mail to:

jonn@microsoft.com

• Text format, 3 1/2" Macintosh 800K disk, please check for viruses

• Text format, 3 1/2" IBM 720K disk, please check for viruses
~~~~~~~~~~~~~~~~~~~~~~~~~~~

• Text format, 5 1/4" IBM 720K disk, please check for viruses

• Text format, 5 1/4" IBM 1.44MB disk, please check for viruses

• Printed, maximum length 50

Limit: Members may submit 1 entry each, 2 if in electronic form. Branch sections may submit 5 entries, 10 if in electronic form. Entries are limited to 50 instructions, 300 if in electronic form. No scatter loading will be supported (instructions must be contiguous).

Legal: All entries become public domain on submission. The ICWS will keep them confidential until after the Tournament.

Tournament Date, Simulator and Location: To be determined. Offers of sponsorship are welcome.

~~~~~~~~~~~~~~~~~~~~~~~~~~~

Just a quick note about Redcode to follow up last issue's discussion of bombing strategies. I am certain many of you thought of this possibility, but I failed to mention it in the last column. Dwarf-like programs can be easily chained together or SPLit to combine different bombing strategies with different goals.

Quarterly Challenge #6

The challenge this issue does not require Redcode but asks an interesting question about modulo numbers. For a core size of 8000, what step size produces the least sum (bomb) free space given 8000 "bombs"? Examples:

0: 7999 + 7999 + ... = 63992000

1: 7999 + 7998 + ... = 31996000

2: 7999 + 7997 + ... + 1 + 1 + 1 ... = 16000001

~~~~~~~~~~~~~~~~~~~~~~~~~~~

Here is my solution to QC#5 Problem 2. The solution to the other problem will appear next issue.

```
; QC#5 Problem 2
; by Mark A. Durham
;
; For core size 8192
SEED DAT          #7
RND  DAT          #0
ONE  DAT          #1
TOP  MOV   SEED,   RND
     ADD   SEED,   RND
     ADD   RND,    RND
     ADD   RND,    SEED
     ADD   ONE,    SEED
     CMP   SEED,  #12
     JMP   HIT
     MOV  #36,     SEED
HIT  MOV   SEED,  @SEED
     JMP   TOP
     END   TOP
```

The Core War Newsletter

Mark A. Durham
Editor

AMRAN
Founding Publisher

William R. Buckley
Founding Editor

The Core War Newsletter, the official newsletter of the International Core War Society since 1987, is published quarterly by the International Core War Society for Core War enthusiasts everywhere.

The entire contents of this issue of *The Core War Newsletter* are Copyright ©1992 International Core War Society. Do not reproduce or distribute without the express permission of the ICWS. Subscription is solely through ICWS membership. Current annual fees are $15.00 drawn on a US bank.

International Core War Society

Jon Newman
Director

Send all correspondence, including renewals and changes of address, to:

ICWS/TCWN
c/o Jon Newman
13824 NE 87th Street
Redmond WA 98052-1959
USA
email: jonn@microsoft.com

Submissions of articles, letters, comments, suggestions, etc. to:

TCWN
c/o Mark A. Durham
18 Honeysuckle Terrace
Spartanburg SC 29307-3760
USA
phone: (803)579-0431
email: durham@cup.portal.com

Back issues of *TCWN* prior to 1992 are available from AMRAN, 5712 Kern Drive, Huntington Beach, CA 92649-4535, USA - email: xwbuckley@fullerton.edu.