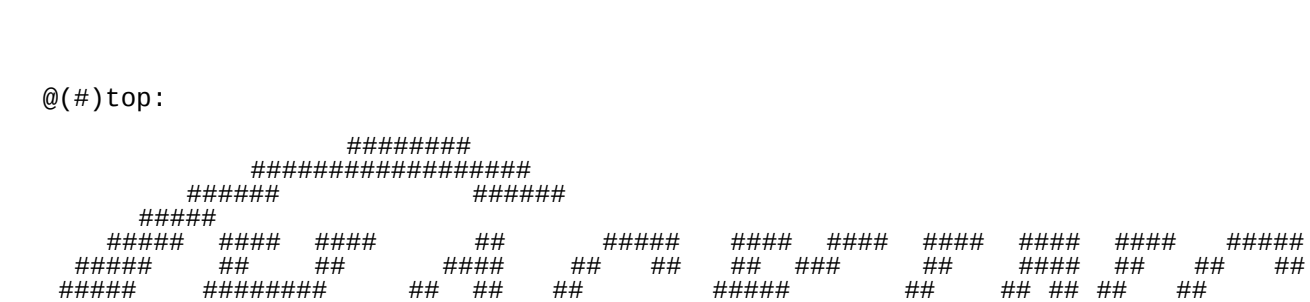
@(#)top:



Issue #13
Version 1.0
July 1996

@(#)contents: Table of Contents

shows how to create a 3-dimensional scene that will bring your dungeon to life. This will give your unsuspecting victim the most realistic gameplay possible.

Departments

1. The (cough, cough) Hacking Editor
   (Reference: editor)
2. Input/Output
   (Reference: io)
3. Newsfront
   (Reference: news)
5. Hacking the Mags
   (Reference: mags)
7. UseNuggets
   (Reference: usenet)
9. FIDO's Nuggets
   (Reference: fido)
11. Hack Surfing
    (Reference: surf)
16. Commodore Trivia
    (Reference: trivia)
17. ? DS, DS$: rem The Error Channel
    (Reference: error)
18. The Next Hack
    (Reference: next)
19. Hacking the Code
    (Reference: code)

@(#)legal: Commodore Hacking Legal Notice

Commodore and the respective Commodore product names are trademarks or registered trademarks of ESCOM GmbH. Commodore Hacking is in no way affiliated with ESCOM GmbH, owners of said trademarks. Commodore Hacking is published 4 times yearly by:

Brain Innovations Inc.
10710 Bruhn Avenue
Bennington, NE 68007

The magazine is published on on-line networks free of charge, and a nominal fee is charged for alternate mediums of transmission.

Permission is granted to re-distribute this "net-magazine" or "e-zine" in its entirety for non-profit use. A charge of no more than US$5.00 may be charged by redistribution parties to cover printed duplication and no more than US$10.00 for other types of duplication to cover duplication and media costs for this publication. If this publications is included in a for-profit compilation, this publication must be alternately available separately or as part of a non-profit compilation.

This publication, in regards to its specific ordering and compilations of various elements, is copyright (c) 1995-96 by Brain Innovations, Incorporated, unless otherwise noted. Each work in this publication retains any and all copyrights pertaining to the individual work's contents. For redistribution rights to individual works, please contact the author of said work or Brain Innovations, Inc.

Brain Innovations, Inc. assumes no responsibility for errors or omissions in editorial, article, or program listing content.

@(#)info: Commodore Hacking Information

Commodore Hacking is published via the Internet 4 times yearly, and is presented in both ISO-8859-1 and HTML versions. This and previous issues can be found at the Commodore Hacking Home Page (http://www.msen.com/~brain/chacking/), as well as via FTP (ftp://ccnga.uwaterloo.ca/pub/cbm/hacking.mag/)

In addition, the Commodore Hacking mail server can be used to retrieve each issue. To request a copy of an issue, please send the following electronic mail message:

To: brain@mail.msen.com
Subject: MAILSERV
Body of Message:

help

```
catalog
send c=hacking13.txt
quit
```

To retrieve a PKZIP 1.01 archive of the individual articles in Commodore Hacking, request the file c=hacking13.zip

To subscribe to the Commodore Hacking and receive new issues as they are published, add the following command to you MAILSERV message prior to the quit command:

```
subscribe c=hacking Firstname Lastname msglen
```

(msglen is largest size of email message in line you can receive. Each line is roughly 50 characters, so 600 lines is about 30000 bytes. When in doubt, choose 600)

example:

```
subscribe c=hacking Jim Brain 600
```

Although no fee is charged for this magazine, donations are gladly accepted from corporate and individual concerns. All moneys will be used to defray any administrative costs, subscribe to publications for review, and compensate the individual authors contributing to this issue.

New: As part of a magazine promotion, Commodore Hacking Issue #12 was professionally laid out on printed format. These printed copies are for sale for US$6.00. Price includes shipping within the US.

Any persons wishing to author articles for inclusion in Commodore Hacking are encouraged to view the submission guidelines on the WWW (http://www.msen.com/~brain/pub/c-hacking-submit.txt) or via the MAILSERV server (send c-hacking-submit.txt).

==========

@(#)rch: Reading C=Hacking

Starting with Issue 11 of Commodore Hacking, the new QuickFind indexing system is utilized to aid readers of the text version in navigating the magazine. At the top of each article or other important place in the magazine, a word prefixed with a special string is present. (See the title of this article for an example. Throughout the magazine, if an article is mentioned, it will be followed by a reference string. For example, if we mentioned this article, we would add (Reference: rch) after the name. By using your favorite editor's search function and searching for the string after the word "Reference:", prefixed by the magic prefix string, will move you directly to the article of choice. To merely skip to the next article in the magazine, search only for the magic prefix string.

Some handy indexing strings possibly not referenced anywhere are:

```
top      top of issue
bottom   bottom of issue
contents table of contents
legal    legal notice
```

For those with access to a UNIX system, the command "what" can be run on the issue, which will result in all the article titles being printed.

A slightly different magic prefix string "@(A)" is used to delimit sub-topics or main heading in articles. The text after the magic string differs depending on article content. For the Input/Output column (Reference: io), the text after the magic prefix will either be "c" for comment, or "r" for response. In features and columns, a number after the prefix indicates the ordinal of that heading or sub-topic in the article. If a specific sub-topic is referenced elsewhere in the article, a sub-topic reference will be indicated. A reference to "@(A)r" would be written as "(SubRef: r)".

As time goes on, the role of this indexing system will be expanded and changed to ease navigation of the text version, but minimize the clutter added by these extra items.

==========

@(#)editor: The Hacking Editor
by Jim Brain (j.brain@ieee.org)

I recently had to choose between my interest in Commodore computers and something else. To many, the choice was clear. Many assured me that hobbies were important, but they simply had to take a back seat when other pressing issues came up. I'll admit that the decision was hard to make. I find that strange, do you? I mean, seriously, it's just an outdated, underpowered, orphaned, incompatible, proprietary, obsolete, 8-bit computer system. Why would I even consider that important?

If you can explain that to me, then you are a true Commodore enthusiast as well. We are all bound together by the immense "pull" of these systems. We don't just "own" them, we treat them like part of the family. We buy toys for them, we help them grow, we accept their limitations, we spend hours with them, and we know everything about them. Although we might have younger and faster family members, we cherish our Commodore. No person or thing could convince us to trade in our familiar family member for a newer, shinier model. As I think of it this way, it seems a bit scary, doesn't it.

Not to leave you in suspense, the "something else" I alluded to above was a new employment opportunity and the subsequent relocation of myself and my family. Even as strong as my feelings are for my beloved machine, I decided that my family came first. Hobbies, no matter how important, are not quite as important. I announced my decision to others who have similar "family members" in their homes, and I pulled the plug on my hobby.

Now, I don't consider myself that important in the scheme of things, but I did underestimate the consequences of my decision. As friends and I tallied up what resources would be unavailable as I left, the amount grew sizable. Luckily, just as with all situations, friends stepped forward to help and keep information from becoming unavailable. Others simply provided moral support and all offered the precious gift of patience while I turned to matters at hand.

I consider myself lucky that so many offered so much to make the situation more tolerable. For reasons unknown to me, it bothered me greatly that deadlines would be missed, pieces of information would go unpublished, important updates would not be updated, and information seekers would find nothing but unanswered questions. Although I knew better, I felt I had deserted the people who depended on me. It's amazing how wrapped up in this I have become.

As you may have guessed, one of the most disturbing resources that was left unfinished was this issue of Commodore Hacking. Although originally scheduled for publication in mid-June, I regretfully shelved it and spent what little time that remained in preparing for a move. Luckily, the move is over, and you now hold the newest issue of this publication.

With this newest issue comes some notes. My wife, Julie, has graciously agreed to offer her services as assistant editor. This will free some of my time to write articles and concentrate on technical article editing. In our quest to find capable writers to author the columns found in each issue, Geoffrey Welsh is now writing "FIDO's Nuggets". We encourage others to help out in this way. Finally, due to the delay in publishing this issue and the length of some submissions, this issue is far larger than our maximum desired size. We apologize for those who will find the excessive size a problem, but the timeliness of the articles and the sheer volume of current events information prevented reduction in size. We will return to a more manageable size by next issue. As well, we created a professionally laid out and printed version of Commodore Hacking Issue #12. If you would like one of these copies, please see "Commodore Hacking Information" (Reference: info) for more information.

Enjoy YOUR magazine,

Jim Brain (j.brain@ieee.org)
editor

==========

@(#)io: Input/Output

Obviously, Commodore Hacking depends on the comments and article submissions from the Commodore community to flourish. Everyone sees the articles, but let's not forget those comments. They are very helpful, and every attempt is made to address concerns in them. Address any comments, concerns, or suggestions to:

Commodore Hacking

10710 Bruhn Avenue
Bennington, NE 68007
j.brain@ieee.org (Internet)

@(A)c: So, You Think You're Fast Enough, Eh?

From: Ralph Mason

Dear C=Hacking,

Keep up the good work with C=Hacking. I was just reading your article about the Super CPU and thought I would add my 2p worth.

You noted that the SuperCPU appeared to be 21.79 times faster but attributed this to the VIC chip stealing cycles. I think this is only part of the story (the smaller part). I think the most cycles are likely to be lost or gained due to the jiffy interrupt routine. The standard 64 executed this routine 660 times and scanned the keyboard etc. during it's count from 1 to 10000. The SuperCPU only executed this code 31 times. Far more of its cycle was spent actually doing work. I think if you could turn off these interrupts you would find that the SuperCPU is actually running short of the 20 times faster than it appears to be showing.

It's almost stooping to silly IBM style Norton SI numbers or other useless benchmarks. These will never show the true story. From what I've read, I'd guess (user's will see) a real world speed enhancement running real application of around 400%, more or less depending on the app.

Cheers,
 -Ralph Mason

@(A)r:
Jim Brain replies,

Ralph, after reading your explanation, I think you are correct in stating that the bulk of the time saved on a 20 MHz unit is indeed due to the fewer interrupts it must service in a given time frame. However, since we can rarely turn off the 60 cycle interrupt, the effective speed is what people will notice. Also, while I think you are correct on this discussion waxing philosophic, I believe most users should see more than 400% increase in applications. Of course, YMMV (Your Mileage May Vary).

@(A)c: A Round of Ice Water for the Editors

From: drankin@crashb.megalith.miami.fl.us (Dave Rankin)

Thank you for all your efforts and putting out this Mag. I and many others do enjoy seeing all this activity for the 8 bit Commodore.

Dave

@(A)r:
Thanks for the letter. We always enjoy knowing that the hours we spend producing this magazine are appreciated by those in the community that read it.

@(A)c: There's Nothing Like the Real Thing, Baby(tm)

From: cjbr@gonix.gonix.com (Jim Lawless)

Dear C=Hacking,

Just wanted to express my enthusiasm for your electronic publication and hope to make regular contributions in the coming months.

I was a C64 hacker from '84 until about '87 when I progressed throughout the Amiga and into the PeeCee world.

I found out about the C64 emulators for MS-DOS/Windows...etc. and downloaded one this morning. It was a great feeling seeing the '64 startup screen again!

My wife expressed some curiosity seeing a pile of old Transactor magazines next to the recliner today. I told her how excited I was about the emulator.

This evening, she returned from a church auction with a C128, a 1541, a 1650 modem, a westridge mode, and a bundle of software all for $30.00.

I guess it's time to get back to my roots and have some fun!

Jim Lawless,
cjbr@gonix.com

@(A)r:
We appreciate the thanks. In addition, we always encourage Commodore enthusiasts to submit articles to the magazine. However, we are most grateful that you have come home again. While emulators have their downside, we have noticed that many who download one end up buying a real machine and rediscover the simple elegance of the Commodore computer. We applaud you for your choice.

@(A)c: Copy Rights!

From: EricJ1@aol.com

I'll make this short and sweet. But, I have to tell you, I love C= Hacking. I'd like to post this as a public bulletin on my BBS if it is not a problem.

Thanks

Eric

@(A)r:
We encourage redistribution of Commodore Hacking for non-profit means. Simply read the guidelines in the issue's legal statement (Reference: legal). As long as the conditions in that guide are met, we would love to see C=H spread throughout the Commodore community.

==========

@(#)news: Newsfront

@(A): ACE Release #15 ACE-15 Programmer's Reference Guide

For those of you who have taken advantage of the Advanced Computing Environment (ACE) operating system written by Craig Bruce, Craig has published the programmer's reference guide for Release #15 of this popular application environment. It is available in the following locations:

ftp://ccnga.uwaterloo.ca/pub/cbm/os/ace/ace15-prg.doc
http://ccnga.uwaterloo.ca/~csbruce/mycommie.html

If you haven't used ACE before, you should give it a try.

@(A): Unscientific Study Proves Commodore Computers are Preferred!

It seems that as homely as some may think the Commodore computers are, children warm up to them very quickly. In fact, the machines are chosen over more expensive machines, as the following stories attest:

James Grubic (grubic@avicom.net) wrote:

> One of the teachers in the school I'm based in actually enjoys using the older computer systems like the Apple IIe, and her students are truly excited about using them. The other day, I gave them a 64c to use, and they were blown away! If you could just see it...a whole gang of youngsters gathered around the C64, waiting for their turn at Jupiter lander...almost brought tears to my eyes.
>
> Needless to say, I'll be arranging for them to get another one.

And Bob Masse followed up with:

> I am not surprised. My little nine year old nephew has a brand new pentium beast with all the goodies, and he is scared to be in his room alone with it when it is on! On the other hand when He comes over to his Uncle Bob's house he has a tantrum to use this old Commodore.
>
> Bob
> kh6zv9@pe.net

So, once again, bigger is not always better!

@(A): Assembly '96 Is Coming!

Have you ever been to a "demo party"? Well, if not, you are missing one of the staples of the Commodore scene since the beginning of the reign of the Commodore computer. Assembly is one such party held in Helsinki, Finland.

In case you aren't aware, demo parties are where demo programmers, computer graphics artists, and computer music artists gather to compete for prizes. Assembly '96 holds parallel competitions for PC, Amiga, and C64 computer systems.

Assembly '96 is to be held August 16 to 18 in the Helsinki Fair Center, Rautatielaisenkatu 3, Finland. Tickets are available for US$50.00. If you are in the vicinity, you should stop by and peruse the 1996 Commodore 64 entries. If, however, you would like to compete in the Commodore 64 class, please read the rules and information packet at: http://stekt.oulu.fi/~mysti/the_sharks/

Prizes of cash are to be awarded to 1st, 2nd, and 3rd place winners in the demo, graphics, and music categories.

For more information, you can contact the organizers via the following ways:

Voice:	ASSEMBLY Org. +358-0-777 3741
WWW:	http://www.assembly.org/assembly96
E-mail:	assembly@assembly.org
IRC:	#asm96
Normal mail:	ASSEMBLY '96 Lakkisepantie 13 00620 Helsinki FINLAND

@(A): Where in the world is Novaterm 9.6 (NovaRom)?

Late last year, Nick Rossi informed the Commodore community that he was developing a new version of his popular 64 terminal emulation software, Novaterm 9.6. However, Nick stated that 9.6 would be marketed as a commercial product, not as a shareware offering as in previous versions. Well, as with all announcements, speculation as to what the new version would include filled up the communication channels for quite a while. Then, in early 1996, the news that Novaterm 9.6 was to be marketed on CARTRIDGE surfaced. Nick cited concerns over piracy and ease of use in deciding to try the cartridge route. Users who asked were told that Novaterm (NovaRom by some accounts) would ONLY be offered as a cartridge.

Performance Peripherals Inc. (PPI) was chosen to manufacture and market the new version. Tentative offering included the basic cartridge and an option that included PPI's CommPort Swiftlink-compatible cartridge and a PPI 3 slot cartridge expansion unit.

Since creating a cartridge requires a higher level of code robustness, delays in the introduction generated reports that Nick was having trouble getting the code to a ROMable state. Other reports mentioned that PPI status as a part time endeavor was the reason for the delays.

Whatever the reason, the following announcement was made by Nick Rossi concerning Novaterm 9.6 on July 5, 1996. Contrary to earlier reports, the software will be available on disk format only and will be initially be marketed directly through Nick Rossi:

NOVATERM 9.6

Bring the telecommunications revolution
to your Commodore 64.

After many delays and headaches, I'm excited to finally announce the release of Novaterm 9.6!

Novaterm 9.6 is available ON DISK, in either 1541 or 1581 format. It comes with a 90-page user's manual. The price for the disk and manual is US$29.95.

ORDERING INFORMATION

Send check or money order for US$29.95 to:

Nick Rossi
10002 Aurora Ave. N. #3353
Seattle, WA 98133 U.S.A.

INTERNET CONTACTS

Check out the Novaterm 9.6 web site for more information:
http://www.eskimo.com/~voyager/novaterm.html

My e-mail address is voyager@eskimo.com.

NOVATERM 9.6 FEATURES

Novaterm 9.6 supports the following new features:

* Zmodem upload, download, auto-download, and crash recovery. Also supports streaming mode with the buffer.
* Ymodem-g and Xmodem-1k-g streaming protocols with the buffer.
* Use any RAM expansion device as the buffer: REU, BBGRam, GEORam, RAMLink or RAMDrive partition, C128 VDC memory.
* "Buffer recovery" feature retains contents of the buffer between Novaterm sessions as long as the memory device does not lose power or get overwritten.
* Text editor can read and write files directly from the buffer.
* Supports the SwiftLink, CommPort, HART cartridge, and Daniel Dallmann's 9600 bps user port enhancement (see http://rpool1.rus.uni-stuttgart.de/~etk10217/proj.html).
* Supports the C128's fast-mode 80-column screen in terminal mode (25, 28, 43, and 50 line modes available).
* C64 80-column emulation features "scroll-ahead" for better scrolling performance. Optionally supports a fast scroll if you have an REU.
* Built-in ASCII translation and UUencode/decode options
* Built-in 80-column file viewer
* Reads real-time clock devices (BBRTC, CMD drives) for terminal mode clock display
* Single-menu loading of terminal emulations (finally!)
* A step-by-step user-friendly configuration utility

Novaterm 9.6 still supports the basic feature set:

* Terminal emulations: ANSI graphics, VT100/102, VT52, Standard, and Commodore graphics in 40 or 80 column mode
* Protocols: Zmodem, Ymodem batch, Ymodem-g, Xmodem-1k, Xmodem-1k-g, Xmodem-CRC, WXmodem, Kermit, Punter, Multi-Punter
* Hardware flow control for high-speed modems
* Script language for automatic operation
* Multiple 19-entry phone books
* 16 user-definable macro keys
* Miniature BBS module / answering service
* Text editor utility with integrated script compiler
* ASCII table editor and Font editor utility

I could keep going, but you get the idea! Novaterm 9.6 supports all of the standard features from previous versions, but its capabilities have been greatly expanded.

Thanks for all the support and suggestions -- the new version finally made it!

@(A): BBS Magazine dead, Long Live Some Trees

Gaelyne Moranec, writer of articles for magazines such as Commodore Hacking (Reference: uqwk), Commodore World, and BBS Magazine, reports that BBS Magazine is no longer. Cited as a magazine for BBS operators and users, the magazine contained a monthly series by Moranec on Commodore BBS users and systems. Being one of the few magazines not Commodore specific to cover Commodore content, its demise is sad indeed. Evidently, the magazine continued on for one issue as _BBS.NET_ but has not been published since. Some of the writers for BBS will be given space in a new magazine to take the place of BBS, but the focus will be on sysops and sysadmins. Gaelyne hopes the new magazine will allow her to continue to write, but she is somewhat doubtful of the prospect.

@(A): Hide the Wolf PC: Little Red Reader-128 2.5 released!

Craig Bruce has released version 2.5 of Little Red Reader-128, the popular freeware utility that allows Commodore 128 owners with 1571, 1581, or CMD FD drives to read IBM PC disks. Features available in the new release include:

* miscellaneous bug fixes
* date support for reading and writing files
* counts of bytes of files in a directory
* remove Commodore files

The program is available from the following locations:

   ftp://ccnga.uwaterloo.ca/pub/cbm/util128/lrr25.uua   (uuencoded archive)
                                                lrr25.doc   (documentation)
                                                lrr25.asm   (assembly source)
   http://ccnga.uwaterloo.ca/~csbruce/mycommie.html

@(A): Basement Boys Software Demise

The geoClub UK newsletter reports that Commodore software developer and distributor Basement Boys Software has ceased operation. Fortunately, Basement Boys Software completed all paid orders and settled all reported business before closing its doors. While we regret the closing due to "lack of support", we are impressed with the ethical methods of doing so.

@(A): LOADSTAR LETTER Going Subscription

As reported in "Hacking the Mags" (Reference: mags), LOADSTAR LETTER will become a subscription based publication. The LETTER, currently bundled with issues of LOADSTAR and LOADSTAR 128, contained 8 pages of additional content not found in either LOADSTAR or LOADSTAR 128. J and F Publishing, which publishes the LOADSTAR line of software and magazines, cites increasing costs and the need for more editorship support in deciding to change the magazine's status from free to subscription. The LETTER will be bundled with the disk magazines until Issue #37. A one year subscription can be purchased for US$12.00 from:

   LOADSTAR Letter
   P.O. Box 30008
   Shreveport LA 71130

Starting with Issue #37, Jeff Jones will join with Scott Eggleston and others to turn the LL into a more hard hitting magazine with fewer ads. The new magazine will continue to run articles by Jim Brain, Gaelyne Moranec, and Jeff Jones, among others. J and F is trying to break 1000 subscribers in order to keep the subscription rate for future subscribers at US$12.00.

@(A): The Commodore Cruiser Is on the InfoHighway

John Brown, of Parsec, Inc., has announced the arrival of the Commodore Cruiser, a subscription based Commodore support BBS system. Accessible via direct phone lines and the Internet, The system is Internet accessible via a telnet to jbee.com. John is offering a free account to each Commodore User Group that requests one. For users, subscription includes full Internet access, as well as Commodore specific areas and file transfer areas. For more information, contact Parsec at:

   JBEE
   Parsec, Inc.
   PO Box 111
   Salem, MA 01970-0111
   USA

@(A): Commodore and Amiga Technology Sold (Again!)

By InfoWorld Staff

  Posted at 3:45 p.m., PT, April 11
  Financially troubled German PC retailer Escom AG said Thursday that it
  will sell its Amiga Technologies GmbH subsidiary to Visual Information
  Services Corp. (VIScorp) of Chicago in a $40 million transaction.
  SEscom acquired the Commodore and Amiga computer technology, patents,
  Sintellectual properties, and brand names in April 1995 for $10 million
  Sat a bankruptcy auction for Commodore International, which filed for
  Sliquidation in 1994. Escom earlier this year itself reported losses of
  S$85 million for 1995, prompting founder Manfred Schmitt to resign last
  Smonth. Selling Amiga will allow Escom to better concentrate on its core
  Sbusiness of PC retailing, Escom said in a statement. VIScorp, which
  Smakes set-top boxes, will acquire the Amiga and Commodore technology and
  Sintellectual property, but not the Commodore brand names, Escom said.

VIScorp is online at: http://www.vistv.com

@(A): DisC=over a New Commodore Specific Technical Magazine

As reviewed in "Hacking the Mags" (Reference: mags), there is a new Commodore publication available. Citing itself as the "The Journal for Commodore Enthusiasts", DisC=overy contains technical content analogous to that found in the defunct Transactor magazine and Commodore Hacking.

Available only in text format, the magazine is available at:

   http://www.eskimo.com/~drray/discovery.html

Alternately, the magazine can be requested via email from:

   s021126@dominic.barry.edu

@(A): CMD SuperCPU unveiled

Initial reports of the CMD SuperCPU are overwhelmingly positive. In fact, it is reported that one European publication would not believe a commissioned review of a beta unit and requested a first hand look at one before they would print the review. Suffice it to say they were impressed as well.

For a report that Guenther Bauer wrote on the new accelerator, check out his review at:

   ftp://ftp.giga.or.at/pub/c64/Super64CPU_test.txt

One of the units traveled to Michigan where Maurice Randall (developer of GeoFAX and owner of Click here Software) debuted it in the US to the Lansing Area Commodore Club. Tim Lewis, LACC President, reported to USENET after the debut:

  "I am one of the few lucky people who have seen for myself what the new Super64 CPU can do. It is nothing short of INCREDIBLE!!!

  For all of you serious GEOS users, I can honestly say this: GET IT! It is money that will not be thrown away! The processing speed is amazing. If you use the Super64 CPU with a REU, I will guarantee you that you cannot go wrong! You have to see it to believe it! Club members that saw Maurice Randall demo this could not believe their Seyes! I was watching this go thru a directory of files, and it just Sflew!

  Folks, you have to see this to believe it! My hats off to CMD, they have really outdone themselves! All I can say is: (sic)COGRATULATIONS!!!"

For more information on CMD or the SuperCPU, contact CMD or visit their WWW Site:

   Creative Micro Designs, Inc.
   P.O. Box 646
   E. Longmeadow, MA 01028
   (413) 525-0023
   http://ww.the-spa.com/cmd/

@(A): Commodore Hacking Contributes to Computer-Mediated Communication Magazine

Following a call for articles in alt.zines on hurdles faced by electronic magazines, Jim Brain contributed an article on the challenges faced by Commodore Hacking. Brain, editor of Commodore hacking, cited the challenges of providing a text version of the magazine for Commodore owners, while attempting to draw out of the closet Commodore enthusiasts online with a hypertext version of the publication. The full text of the published article is available at:

   http://www.december.com/cmc/mag/1996/may/brain.html

@(A): "Zelch" Down for the Count

In C=Hacking #12, we noted that Bo Zimmerman had connected his Commodore 128 to the Internet, albeit through a Linux system. Well, as all good things must end, Bo has taken down the BBS system due to hardware overheating problems. However, Bo hopes to provide documentation on how the system was set up so that others can configure similar systems.

@(A): The "Official" DesTerm WWW Site

In March, Matt Desmond, creator of the popular 128 terminal emulation program DesTerm, announced that he is now online at:

   http://www.ionline.net/~mdesmond

It contains information about Matt, but is more importantly the gateway to the "Official DesTerm Page." The site contains information about the

new 3.0 version of DesTerm that Matt is developing.

@(A): Compuserve INformation Service = Compuserve Internet

On May 21, Compuserve (CIS) announced it would phase out its proprietary software and services in favor of providing service using Internet standards. The company hopes to re-launch itself as an Internet provider by year's end. The new service will be accessible through a standard World Wide Web browser. It is unclear how this change will affect Commodore users who rely on Compuserve's "shell" access for Internet and Compuserve specific access.

@(A): Creative Micro Designs, Inc. New Sponsor of Genie CBM RTC

Creative Micro Designs, Inc., has taken over as the sponsor of the Commodore RTC area on Genie. The Commodore RTC remains one of the few well utilized places to stay current on Commodore events and find Commodore information. CMD cited an interest in providing quality information for Commodore enthusiasts as a driving reason behind the decision to sponsor the Genie forum.

@(A): Hail the New Prez

Meeting 64/128 Users Through the Mail, a non-profit organization designed to allow Commodore users to unite and gather information about their machines via mail, has announced a change in presidency:

The new president is Tom Adams, and the new address for club correspondence is as follows:

> Meeting 64/128 Users Through the Mail
> c/o Tom Adams, President
> tom.adams@neteast.com
> 4427 39th St.
> Brentwood, MD 20722-1022

If you are interested in membership, please contact Tom. The club is especially useful for those who live in areas with no Commodore support.

@(A): Commodore VIC-20 Newsletter Address Change

For those interested in the Commodore VIC-20, a very useful but under utilized computer, Jeffrey Daniels publishes a newsletter for the machine. The publication address has changed to:

> Vic Newsletter
> Jeff's Ink Press & Deli
> P.O. Box 477493
> Chicago, IL 60647 USA
> Jeffrey Daniels, editor
> U17632@UICVM.CC.UIC.EDU

A copy can be obtained by writing the above address.

@(A): ESCOM Does a CBM! (Well, Not Really)

Financial Time/Edupage: July 4, 1996

> "Escom, the German company that is one of Europe's largest PC retailers, is seeking protection from its creditors (similar to Chapter 11 protection in the U.S.), following significant trading losses, and losses caused by a stock write-down. Aggressive expansion into new markets such as the U.K. had caused storage and supply problems."

Since ESCOM had recently sold the rights to the Commodore and Amiga lines to VISCorp, the filing will have little affect on Commodore 8-bit owners. Also, CMD reports that this action is part of a massive reorganization effort by ESCOM intended to solidify its PC manufacturing operation. CMD notes that, unlike CBM, ESCOM is NOT liquidating, but merely employing a common US business tactic of filing to shield themselves from creditors while reorganinzing the business.

==========

@(#)trick: HEAVY MATH - Part 0: History, Arithmetic, and Simple Algorithms
by Alan Jones (alan.jones@qcs.org)

Someone on comp.sys.cbm asked if the C64 could do HEAVY MATH, meaning solve computationally intensive numerical problems. The answer is of course, YES! This is the first of a series of articles on numerical computing for the C64/128.

@(A): Introduction

The C64 is not the best computer for numerical work. However, it does quite well within its limitations of speed and memory. It is fine for most homework and hobby related problems, but not for big industrial problems. It does not bother me at all to let it crunch numbers while I watch a movie or sleep. Those old commercials about sending your children to college with a C64 were a joke. Still, it can save you a long walk to the campus on a miserable night. And you can always use it as a terminal to check jobs running on the mainframe.

The C64 is also a good computer for developing numerical algorithms and programs. You can try new ideas and write programs at your leisure at home with a C64. When developed to your satisfaction, algorithms and programs can be "ported" to bigger and faster computers to solve larger problems. The C64 has many programming languages available, although many are not well suited for numerical development work. On larger computers Fortran and C are popular for numerical work. On a C64, Power C might be a good choice for some users. I use COMAL 2.0. I also have COMAL programs that can help convert source codes from BASIC to COMAL, and COMAL to Fortran.

Our C64 with its 6502 (6510) and 64K of RAM is a very simple machine. It is so simple that many contemporary numerical programs are far from ideal on a C64. So I will start with a bit of numerical computing history. Early computers and the numerical algorithms that they used are often closer to ideal for the C64 than contemporary PCs. Researching old numerical algorithms can be useful for the C64; e.g. Quartersolve in C-Hacking #10. Of course new algorithms are useful also and sometimes you might want to combine ideas from both sides of the spectrum.

@(A): History

In the beginning... were fingers. Seriously, "computer" was a human job description. These days, human computers are just an oddity seen on TV talk shows. The invention of logarithms was a big boon, and log tables and slide rules were just the start of computational aids. Eventually, mechanical adding machines were developed for high precision, error free (but slow) numerical work. One can still find large desk top Friden and Monroe mechanical adding machines. Numerical work was still a slow tedious process. More computing tools were developed. The Differential Analyzer was a mechanical computer that could solve IVPs (Initial Value Problems, integrating differential equations). There were also some early analog electronic computing aids. The first electronic analog computer was actually developed after electronic digital computers. (One could argue that many WW II autopilots and automatic control circuits were electronic analog computers.)

The first digital electronic computers were the ABC, ENIAC, EDVAC, and UNIBLAB. (UNIBLAB is just for the Jetson's fans. ;)) John Vincent Atanasoff invented the first digital electronic computer at Iowa State University. (So if someone answers the phone and says, "He's on the John. Can he call you back later?" It might not be mean what you first think.) Clifford Berry, was a grad student and chief technician, hence the Atanasoff-Berry Computer, or ABC. The Atanasoff story is fascinating. See: The First Electronic Computer: The Atanasoff Story, Alice R. and Arthur W. Burks, The University of Michigan Press, 1988.

Atanasoff wanted to be able to solve large sets of linear equations. Even with large mechanical adding machines, solving a 10 by 10 problem was about the largest size that would be attempted. Schemes to connect several mechanical adding machines were not feasible, and analog devices were not precise enough. He was working at a small university and the small grants available to him were a serious constraint. He developed the ABC over a couple years for less than $7,000. The ENIAC would later cost about $500,000! Atanasoff invented a way to use electronic vacuum tubes as high speed digital switching devices. He then invented a serial arithmetic logic unit, ALU. Vacuum tubes were still too expensive so he used cheap capacitors for memory. He invented additional circuitry to refresh the capacitors, i.e. dynamic RAM. He designed a parallel computing machine that could add (and subtract, shift, NOR,...) 30 50-bit binary numbers using 30 modular ALU units. This allowed it to solve up to 29 linear equations with one right hand side vector. The design could easily be scaled up in size and precision. It used scratch paper for I/O and temporary memory. (Created in man's image?) The card punch/reader was the limiting factor. Mechanical punches, like (then) new accounting machines might use, were too slow. An electronic spark punch was developed. A dielectric material (paper) was placed between electrodes. A high electrical

voltage would carbonize a dot in the material and actually burn a small pin hole. A smaller voltage would later test for the mark. This was actually Berry's project. It had decimal to binary and binary to decimal conversion for initial and final I/O, as well as other nice touches.

Atanasoff also developed a variation of Gaussian elimination for solving the linear systems of equations with the ABC. The ABC, like our 6502, has no multiply instruction. The ABC had capacitor memory to hold two rows of equations. Multiplication was done with shifts and adds, but whole rows were computed in parallel. Fixed point binary arithmetic with truncation (no rounding) was used. However, it provided 50 binary bits of precision which was more than the adding machines provided. It used no division. The result would be printed (punched) out in decimal as two integers that would be divided on a mechanical desk calculator for each variable. His numerical algorithm may be useful for our 6502, although I'm sticking with the slower floating point arithmetic. It was not a general purpose "stored program" computer, but it could have been adapted to solve a variety of problems.

The ABC was completed and operational in April or May of 1942 except for one problem: The card punch reading was not reliable. The problem may have been the dielectric material or choice of paper. A 5 by 5 problem could be reliably solved, but not the larger problems that it was designed for. The problem could have been fixed. However, Atanasoff and Berry were called to other WW II related work and not allowed to perfect the ABC. The ABC was stored and later dismantled. Ironically, the war that built the ENIAC killed the ABC. Of course many of John Atanasoff's original inventions were later used in the ENIAC and EDVAC computers.

The ABC was built into a desk sized wheeled cart and could be transported to a researcher's "home." It cost less than $7000, but additional units would have been cheaper. The ABC was akin to our favorite low cost home computer. By contrast, the second computer, ENIAC, cost a fortune, required a team of technicians to operate, and filled a large room. The ENIAC led to monolithic computing centers. It would be decades before the computer returned to the home.

I'll skip the better known history lessons: transistor > microprocessor > electronic hand calculators > home computers > C64 >... And of course the electronic computer caused an explosion in the development of mathematics and numerical algorithms.

@(A): Arithmetic

Arithmetic is the basic building block of numerical algorithms. There are many types of numerical variables and arithmetics. Binary arithmetic is the most efficient for intensive numerical work. Decimal arithmetic is best for simple math where conversion to and from binary would just slow down entering and outputting numbers. Floating point arithmetic is easy to use because it is self scaling and covers a large dynamic range, but it tends to be slow. Fixed point, e.g. integer, arithmetic is fast but not as easy to use. Interval arithmetic involves computing not just a rounded result but an upper and lower bound on the result to cover the interval of the arguments and the accuracy of the computation. PGP encryption uses a high precision modular arithmetic. Complex, quaternian, and vector arithmetic can also be used.

The C64 built in BASIC provides 5 byte floating point variables and arithmetic and 2 byte integer variables. I think integer arithmetic is done by converting to floating point. Most of the programming languages for the C64 use the same numerical variable types and even the same arithmetic code. Even in assembly language we often call the same floating point arithmetic routines. The +, -, *, and / arithmetic operations on the C64 have no bugs. However, they appear to be coded for minimum code size rather than minimum execution time. Every type of computer arithmetic can be built up from the 6502 instruction set. Some arithmetics can be coded for specific applications such as Polygonamy in C-Hacking #12.

My interest is in using the floating point routines with numerical algorithms and writing programs. Of course even floating point arithmetic routines are built up from smaller arithmetic blocks. The key building block is the multiplication of two positive 8 bit values into a 16 bit result. Our 6502 has no such instruction.

The 6502 CPU was designed to be a low cost 8 bit CPU. It is fairly cheap to interface to and will quickly access cheap "slow" memory. It is also very quick and responsive to interrupts. It can perform 8 bit binary and BCD addition with carry. The Z80 CPU was designed to be the

ultimate 8 bit CPU. It has several 8 bit internal registers which can be used in 16 bit pairs. It has a full instruction set that includes some nibble oriented instructions and a 16 bit add. On average a 1 Mhz 6502 is about as effective as a 2 Mhz Z80, and Z80s are generally available in faster speeds. The C128 has a Z80 CPU that could be used for numerical work, but it was poorly integrated into the C128 and offers us no advantage over the 6502 (other than executing CP/M and other Z80 code). Neither CPU has a multiply instruction. The fastest way to multiply with a Z80 is with the simple binary shift and add method. However, this is not true with the 6502! The fastest way to do math on a 6502 is by using table look ups. This opens the door for creative programming solutions.

Tables can use up a lot of memory, especially for a function of two or more arguments. An 8 bit multiply table could eat up 128K of memory. A 4 bit, or nybble, multiply table would only need 256 bytes, but this would involve so much additional work to realize the 8 bit multiply that it is hardly worthwhile. The C64/128 multiplies with the slow binary shift and add method. However, it is not so slow that we can use disk or REU memory to speed up such a simple function (a large bank switched ROM would be much faster). The table look up method can be readily used when multiplying by a constant, such as when calculating CRCs. Now consider the algebraic identity,

$$a*b = ((a + b)/2)_2 - ((a - b)/2)_2.$$

With some more work we can do the multiplication using a table of squares of only about 512 bytes! (a + b) could overflow to nine bits, but we will immediately shift right one bit (the division by 2) so this is no problem. However, if (a + b) is odd the least significant bit is lost. This is easy to test for by doing a Roll Right instead of a shift and testing the carry bit. One way to compensate is to decrement a by 1 (a <> 0), multiply as above and add b, a*b = (a-1)*b + b. The decrement is free, but we pay for the extra add. Using 256K of external memory you could do a 16 bit multiply this way.

For an example of the shift and add type multiply and divide see, "High-Speed Integer Multiplies and Divides", Donald A. Branson, The Transactor, Vol. 8 , No. 1, July 1987, pp. 42-43, 45. Note also that although a*b = b*a, the ordering of the arguments can effect the multiplication speed depending on the bit patterns.

Perhaps a year ago there was a discussion running in comp.sys.cbm on ML routines to do fast multiplication. There was no clear best solution. Performance often depended on where the arguments a and b were and where the product was to be stored. This also affects how well these building blocks can be used to perform multi byte arithmetic.

Division is a more difficult problem. It can be done by shifting and subtracting, table look up, and algorithms based on computing the inverse. Consider: a/b = exp(log(a) - log(b)). With tables of the logarithm and exponential functions (and you might want to use base 2) we can do division with three table look ups and one subtraction. The log and exp functions will have to be tabulated to a greater precision than the arguments and result, or it will only produce an approximation. In most cases we will still have to calculate the remainder using multiplication and subtraction. Of course with log and exp tabulated we can calculate fast approximations to many other functions, including multiplication.

Stephen Judd used multiplication based on a table of squares and division based on a table of log and exp in Polygonamy in C-hacking #12. He reported that his 9 bit/8 bit divide takes 52 cycles "best case." However, where numerical algorithms are concerned, only worst case and average case performance are important.

Double precision, and multiple precision arithmetic routines should be coded efficiently in assembly language using the fast building blocks suggested above. However double precision FP variables and arithmetic can be built using pairs of ordinary FP variables and arithmetic. This will be slow but it can be effective when used sparingly such as when testing single precision algorithms or using iterative improvement techniques. See, "Double Precision Math", Alan Jones, Comal Today, Issue 20, Feb. 1988, pp. 18-20, and Comal Today, Issue 22, May 1988, pp. 58-61.

@(A): Numerical Algorithms

An algorithm is a procedure or set of instructions for computing something. I am mainly concerned with HEAVY MATH algorithms, but here I will present only feather weight numerical algorithms.

Consider the trivial algorithm,

```
repeat
   x := (x + 1/x)/2
until converged
```

This is a stable quadratically convergent algorithm. For any initial x <> 0 it will converge to sign(x), i.e. +1 or -1. Pick a number, say 1.5 and take a few iterations. Note how fast it converges to 1.0. The error or distance from 1 keeps getting squared down toward zero. The number of correct digits in each iteration doubles. This is the quadratic convergence. Pick another number such as 10_20 and try again. At each iteration the error is cut in half. We take giant strides but convergence is still painfully slow. This is a linear convergence rate. This is a typical Newton's method algorithm. Near the solution, inside the region of quadratic convergence, convergence is very fast. Outside the region convergence is much slower. On more complex problems convergence may fail altogether or converge to an undesired point. In general an algorithm will converge to a "limit point" and if the algorithm is numerically stable, the limit point will be very close to the exact solution intended. Although it looks like this algorithm could run forever like an infinite series, in finite precision arithmetic it always converges in a finite number of iterations, even from the bad starting points. This algorithm is not so trivial when applied to a square matrix (with no eigenvalues on the imaginary axis). It will compute the matrix sign function which can be used to compute the stable invariant subspace, which can be used to solve the algebraic matrix Ricatti equation, which can solve two point boundary value problems, and be used to solve linear optimal control problems. Not to mention other pseudo random buzz mumble...

@(A): Inverse and Division

The inverse x = 1/b can be iteratively computed from x := x*(2 - b*x). This is best used as a floating point, or multiple byte algorithm. This is a quadratically convergent algorithm. This means that each iteration should double the number of correct bits in x. You could use an 8 bit multiply and converge to an 8 bit solution from an initial guess. A better use would be to compute a 32 bit result (our floating point mantissa). We might start with an 8 bit estimate from x := exp(-log(b)) using look up tables, take an iteration using 16 bit multiplication (or 16 by 8) to get a 16 bit estimate, and take another iteration using 32 bit multiplication to get the final 32 bit result. Division can then be accomplished as a/b := a*(1/b). Of course this is only useful if you have fast multiplication.

@(A): Square Roots

BASIC 2.0 calculates square roots from x = exp(0.5*log(a)). This is slow since BASIC calculates the log and exp functions, and inaccurate as well. If you have these functions tabulated you might want to use them for an initial estimate of x. If you have a table of squares, the inverse function of the square root, you could use a search routine on the table. Square roots can be calculated iteratively from the Newton's method algorithm,

```
x := (x + a/x)/2
```

One can also compute x = 1/SQR(a) using

```
x := x*(3-a*x*x)/2
```

avoiding the division.

E. J. Schmahl published ML code for computing the square root in "Faster Square Root For The Commodore 64" in The Transactor, Vol. 8, No. 1, July 1987, pp. 34-35. This used a 16 byte look up table to start, followed by Newton's method. He called the ROM FP routines to do the calculations, but variable precision arithmetic could also be used as suggested for the inverse algorithm.

Another interesting algorithm for the INTEGER square root was recently published by Peter Heinrich, "Fast Integer Square Root", Dr. Dobb's Journal, #246, April 1996. This is a fast algorithm that uses no multiplication or division. It is not known yet if this is a good algorithm for the 6502.

@(A): Algebraic Geometric Mean

The AG Mean is our first real numerical algorithm, the others above are

our arithmetic building blocks.

```
   Repeat
      a(i+1) := (a(i) + b(i))/2
      b(i+1) := SQR(a(i)*b(i))
   until converged
```

For 0 < a(0) <= 1 and 0 < b(0) <= 1 the sequences converge quadratically to their common limit point, the AG mean of a(0), b(0). Note that we need to use full precision from the start and an accurate square root routine. The BASIC 2.0 SQR routine is not accurate enough. This can be used to compute the complete elliptic integral of the first kind, K(k). With a(0) = 0 ,and b(0) = SQR(1-k_2), K(k) = PI/(2*a(n)). The AG Mean can also be used for some other computations

@(A): A Caution

Many mathematical equations can be found in math books and similar sources. However, these are often in a form for ease of typesetting and further algebraic manipulation. They should not generally be coded as written. For example, the well known quadratic equation is the best way to compute the roots of a second order polynomial equation. However, there is a particular way to code it to avoid overflow, underflow, and loss of precision. There are also analytical expressions for the roots of third and fourth order polynomial equations. However, roots of third and higher order polynomials are best solved for using general root finding techniques.

@(A): Conclusion

This article is long on discussion and short on usable code. Although it suggests faster ways of performing arithmetic on a C64, the built in FP +, -, *, and / routines are reliable and can used for serious computations. If I continue this series, I would want each article to present source code for solving a numerically intensive problem. In Part 1, I present an introduction to Linear Programming. Hopefully other topics will be suggested by readers, and possibly articles will even be written by other users. Of course I could also write articles on numerical methods, or turn this into a simple question and answer column. I suspect many readers have already written many HEAVY MATH C64/128 programs but have not shared them with the Commodore user community yet.

==========

@(#)mags: Hacking the Mags

Not everything good and/or technical comes from Commodore Hacking, which is as it should be. (We still think we have the most, though...) Thus, let's spotlight some good and/or technical reading from the other Commodore publications.

If you know of a magazine that you would like to see summarized here, let C=Hacking know about it. These summaries are only limited by Commodore Hacking's inability to purchase subscriptions to all the Commodore publications available. We are very grateful to those publications that send complimentary copies of their publications for review.

@(A): Commodore Gazette

This new introduction is published by Commodore Gazette Publications, and is NOT related to COMPUTE's Gazette, in case you are wondering. In Volume 1, Number 7, editor Christopher Ryan mentions the above fact, as it seems some upset COMPUTE'S Gazette subscribers were calling him. In this issue, you will find some detailed instructions on installing CMD's JiffyDOS, as well as how to turn your 64 computer into a 128 (I should mention this was the April issue). Kenneth Barsky provides some handy tips for BASIC programmers, including one involving the append mode of CBM disk drives. Overall, the fare is a bit light, but is pleasing.

@(A): Commodore World (http://www.the-spa.com/cmd/cwhome.html)

In the continuing saga of the funky graphics, Jenifer Esile, who made a good share of them, has resigned from editorship of Commodore World. We hope it isn't something we said :-). Anyway, CW has hired a new assistant editor, and two new issues have rolled off the press.

Doug Cotton, the editor of CW, mentioned that Issue 13 was a nightmare. I guess even CMD falls prey to the superstitious number. No matter. For those wanting to learn more about the World Wide Web and HTML, Katherine Nelson presents an article on how to use this presentation markup language to develop exciting WWW sites. A glimpse of the

Commodore LCD computer is given, and Doug Cotton presents his RUN64 loader, also presented in the last issue of C=H. For those who are anticipating the new release of Novaterm, Gaelyne Moranec interviews Nick Rossi, the author of Novaterm.

Issue 14 follows up on the HTML tutorial by Katherine Nelson. Since Commodore software is developed on many computer platforms, Doug Cotton presents an article on transferring files between dissimilar computer systems. In the reference department, clip out the User Group list compiled in this issue. Obviously, you don't need it, but it's something to send the clueless person who calls asking for help. Jim Butterfield shows how to get some input into your ML programs, and Maurice Randall delved into the VLIR file format used in GEOS.

@(A): DisC=overy (http://www.eskimo.com/~drray/discovery.html)
Subtitled "The Journal of the Commodore Enthusiast," this recent publication introduction debuted online on May 17. Available in electronic format, like C=H, this is a magazine Commodore Hacking readers won't want to miss. Issue #1 includes articles by Stephen Judd on VDC timing, by Nate Dannenburg on constructing an 8-bit analog to digital board, and by Mike Gordillo on upgrading the 16kB 128 VDC to 64kB. Other articles include a discussion on George Taylor's new Tri-FLI technique, an overview of CP/M, and a look at ModPlay 128. Commented source is included for many of the articles, and the technical details are not spared. The layout is similar to early issues of Commodore Hacking, but more attention is paid to consistency throughout the issue. In addition to the issue itself, there is a WWW Site devoted to the magazine: (http://www.eskimo.com/~drray/discovery.html). Still uncertain here at Hacking Headquarters is the publication cycle for this new arrival, but we hope it finds an eager audience. The editors are certain that there is room in the Commodore publication arena for DisC=overy and more magazines like it.

@(A): Driven (http://soho.ios.com/~coolhnd/)
Issue #13 contains a good review of the 1541-DOS package from Bonestripper. For those who don't know, 1541-DOS allows your 1541 to read and write a disk format that can be read on IBM 5.25" floppies. Iceball presents a reality-check for the demo scene, while Tao discusses some ideas to help developers write graphics-format independent code. Even if you don't develop graphics code, you should read this article and heed its warnings. Failing to test NTSC code on PAL machines or vice versa can impact the usefulness of your application. A little extra effort in development can pay off in the end. Finally, Tron presents some more information on Internet Relay Chat (IRC), including how to use its features.

Eclipsing the last issue, Drive #14 offers a wealth of information. Nate Dannenburg presents information on ModPlayer 128, while Guenther Bauer reviews the new CMD 20 MHz SuperCPU accelerator. Nate describes some of the theory behind creating digital music and how it can be done using a Commodore 64. Lastly, Issue #14 presents a transcript of the Genie roundtable discussion on the 64 and its place on the Internet.

@(A): LOADSTAR (http://www.loadstar.com)
Issue 142 brings us Fender's proposal for dealing with the glut of good software languishing in the closets of those who have forgotten it sits there. Adam Vardy presents a screen saver appropriately described as "a screen saver for a computer that doesn't need one." Of special mention on this issue is Terry Flynn's SYSARCH, a handy 14 screen reference guide containing PRG info at the touch of a key or two. For those who have flipped through the 64 PRG enough to wear out the binder, this might provide some relief.

In Issue 143, Jeff Jones presents the nuts and bolts behind LOADSTAR's text packing routines, while CodeQuest '95 silver medal winner Paul Clark offers a handy LIST wedge that allows forward and backward BASIC listing scrolls. Paul's wedge even allows searching. That's a neat twist for you BASIC programers. For those who don't regularly use GEOS but are given graphics in GEOPaint format, Saimak Ansari provides a utility that will allow you to view and print them without GEOS.

By far the most technical of the 3 reviewed, issue 144 contains a number of helpful utilities. One, called Menu Toolbox II, allows the programmer to create useful and functional user interfaces with a minimum of effort. Jeff Jones, the author, has rolled an extensive list of user interface controls into this package. Additionally, Ken Robinson presents some bug fixes and enhancements to Jeff Jones' Static Array System, a package that allows programmers to treat RAM like a relative file.

@(A): LOADSTAR 128 (http://www.loadstar.com)
For all the Dave's Term folks, Issue 31 presents the 5th and final installment of the 128 terminal program. Bob Markland presents his RANDOM 2-254 program that one can use to create random numbers. In addition, Bob presents RLE 128, a utility to Run Length Encode (RLE) fines to make them smaller. RLE packing is especially useful for text screens and other files with repeating symbols. Fender Tucker notes in the introduction that many new 128 titles are arriving for publication, and he mentions that Mr. Markland will be taking charge of more aspects of this publication. We hope he enjoys it.

@(A): LOADSTAR LETTER (http://www.loadstar.com)
We have decided to break LL out from the LOADSTAR reviews because J and F Publishing has recently decided to make LL a separate product. The details are in LL Issue #34. The publication will continue to be free of charge until #37.

In LL #32, LOADSTAR introduces two more editions in its "Compleat" line. The Compleat Crossword offers what the name inplies, while The Compleat Jon presents 11 previously published Jon Mattson games in one compilation. Jeff details a particlularly nasty bug that he worked around in The Compleat Crossword. He invites savvy folks to figure out the problem. In the reference department, most will want to archive Jeff Jones' Introduction to Machine Language. Oh sure, it won't teach YOU anything new, but the tables are sure nice to have if, perchance, a friend ever forgets the addressing modes for some opcode. Lastly, Jim Brain presents part 5 of the Internet series.

LL #33 showed up with a revamped look. The publication now has a professional front splash graphic, and the style has evolved. We are impressed with the new look. Of notable mention is the preliminary information on the CMD SuperCPU and its compatibility. A discussion of BASIC compiler pitfalls and problems follows. Every programer should read and re-read the article on how to write applications that work on machines with "old" ROMs. The problems are so simple, but neglicting them ruins a perfectly fine app on an old 64. If you haven't figured out how to access RAM under ROM and I/O at $D000, there's some functions in the issue to do that as well.

In LL #34, we learn the new email address for LOADSTAR email: jeff@loadstar.com. The issue also mentions LOADSTAR's WWW address: http://www.loadstar.com and notes that it will be the "coolest C64 site on earth." Well, we'll see about that, but credit is due for the attempt. In this issue, LOADSTAR notes the impending change of LL from free to subscription based, and some more information on the SuperCPU is related. For those in the demo scene, you'll be pleased to know that Driven will now be distributed on the 3.5" version of LOADSTAR. Gaelyne Moranec and her WWW site is spotlighted, but the most newsworthy information in this issue is the mention that Byte magazine recently recognized the 6502, the SID, and the Agnes/Denise/Paula chips as some of the 20 most influential ICs in the computer industry.

Although LL will appeal to the beginner to intermediate Commodore user with current events information, we are pleased to see numerous code fragments and technical discussions interspersed with the lighter fare. For $12.00 a year, don't pass it over without a look.

@(A): The Underground
Commodore Hacking would like to thank the anonymous Underground reader who donated a subscription so that we can review this magazine for our readers. We appreciate the donation.

With our first issue, Scott Eggleston has changed the format of the publication a bit. Citing problems with reproduction of the smaller format and printing woes, The Underground gains a whole new larger format look with Issue 13. For those developers considering a CMD hard drive purchase, Disk Estel reviews an HD-40. Two Internet related articles surface in this issue, as Mark Murphy explains some of the technology of the Internet, while Disk Trissel details the File Transfer Protocol (FTP). A full complement of columns and departments accompany each issue as well. The Underground covers beginner to intermediate material and uses GEOS to publish each issue. Digitized photos make frequent appearances, and the content is top-notch.

Other magazines not covered in this rundown include:

* _64'er_
* _Atta Bitar_ (_8 bitter_)
+ _Bonkers_
+ Coder's World

+ _COIN!_
o _Commodore 64/128 Power User Newsletter (CPU)
o _COMMODORE CEE_
* _Commodore Network_
* _Commodore Zone_
* _Gatekeeper_
o _Vision_

Notes on Legend:
* = We have never received an issue of this publication.
o = We have not received a new issue of this publication to review.
+ = We will begin reviewing this magazine in the next issue.

In addition, others exist that C=Hacking is simply not aware of. As soon as we can snag a copy of any of these, or get the foreign language ones in English :-), we will give you the scoop on them.

==========

@(#)os: OS/A65 - a Multitasking/Multithreading Operating System for 6502 Computers
by Andre Fachat (a.fachat@physik.tu-chemnitz.de)
http://www.tu-chemnitz.de/~fachat

@(A): Introduction

In 1989, I first thought about building a self-designed computer. I already had some experience with 6502 based computers. A friend of mine and I had been trying to build a telephone line switch computer based on the 6502. Although the project never succeeded (well, to a certain extent it worked, but then we always got new ideas...), the project gave me an idea of what an OS should be capable of.

With my homebrew computer, I not only wanted to implement one of those 'simple' OSes as in the C64 or other 6502 based computer, but I also wanted to go a step further and do a real multitasking, microkernel design OS. This constrained the hardware design to allow memory mapping of key memory locations, including the 6502 zero-page and stack.

@(A): What Should a Real OS Do?

A real operating system has four major parts that handle the input/output, filesystems, memory management and process handling. At the very least, a "real" OS includes some form of multitasking :-)

Process management forms one block of an OS. A multitasking operating system requires more administration than a single-tasking OS. A process, or task, can be seen as a set of allocated resources. These resources include memory pages, swap pages, open files, and even the CPU, if the task is active. The CPU is the processing element that executes the given program using the allocated resources. Therefore, the CPU state has to be saved if a task is interrupted. This allows undisturbed continuation after the interruption is handled. For each task, the allocated resources have to be registered and freed. As the CPU can be allocated to only a single task at a given time, it must be shared among all the active processes. So, in order to create the illusion of executing multiple processes at the same time (pseudo-parallelism), the CPU has to be assigned to one task after another, at a speed that achieves this illusion. If the assignments happen too slow, the illusion is lost, but if the speed is too fast, the CPU spends all of its time administering the tasks and not enough time executing the tasks. The same concepts hold true for multiprocessor computers, except that such a machine can achieve parallel operation on as many tasks as there are CPUs in the system.

A scheduler interrupts the CPU after a certain time to allow the CPU to be assigned to another task. If the scheduler interrupts the task itself to schedule a new task, the system is called preemptive. If the task has to give the CPU back to the system, it is called cooperative multitasking, like in MS Windows (tm). of the two, preemptive is preferred, as cooperative multitasking fails when a single process forgets or is unable to relinquish control of the CPU. If such a scenario occurs, the computer is "blocked".

As the second part, I/O provides a uniform interface to all peripherals, including character devices (serial lines, parallel printers), or block devices (disk drives). These services are normally provided by device drivers, which, in some operating systems, are even loadable. One problem is the communication between device interrupt routines and the rest of the system. Andrew Tanenbaum, in _Operating Systems, Design and Implementation , says that, "Interrupts are an unpleasant fact of life.

They should be hidden away, deep in the bowels of the system, so that as little of the system as possible knows about them." Nevertheless, interrupts are necessary to handle time critical operations, like providing new data to serial lines. Provisions must be taken to avoid data corruption by an interrupt routine and a program (or the kernel) using the same memory locations at the same time. So, even if you don't like interrupts, you have to use them.

As the third part, the filesystem provides user-level abstraction of I/O. Files store information of any kind. It is the most visible part of the OS. The naming conventions make a big part of the OS view for the normal user. (Remember the 8+3 character filename length restriction in MS-DOS filesystems?) The filesystem itself provides a standard interface to the user, although the underlying structure (i.e. how files are stored) may differ on different devices. In UNIX operating systems, even devices can be used as files and are represented by special entries in the directory structure (on the newest version of Linux (pre2.0.) even files can be used as filesystem (that hold files that can be used as filesystem (that hold files.. Ooops ;-))). I will not go further into this issue, but how a filesystem is organized can sometimes become a religious war among their respective followers. Since a filesystem keeps all internal structures to itself, it is possible to mount differently structured filesystems in one system.

As the final part, memory management keeps track of which parts of the memory are in use and which are not. Memory can be allocated when needed and is freed for other uses when no longer needed. Modern systems use the concept of virtual memory. Virtual memory specifies a system that uses a translation table between the CPU and the real memory locations. When the CPU tries to access a certain memory address, the address given in the opcode does not reflect the real address used to access the memory chips. Instead, the translation table is used to look up the real memory address from the `virtual' address given in the opcode. So, if there is no appropriately sized contiguous memory block available in real memory, such a block can be built using smaller chunks by setting up the translation table for the task. The lookup is done by the memory management unit (MMU). Software called a memory mapper is used to load and change the table. It loads the table with the values set up for each task. So the same opcode address in two different tasks accesses very different memory locations in the RAM.

More sophisticated memory managers even do swapping. The memory manager allows a task to allocate more memory than actually available. If a memory location that is not available is accessed, the CPU is trapped (the ability to do this cleanly was one of the (IMHO very few) additions from the Motorola 68000 to the 68010 CPU). The memory manager then saves (swaps out) another memory page to disk and uses the now free memory. The CPU can then continue. If a swapped out memory address is accessed, the CPU is halted again and the page is swapped in again - swapping out another page if necessary. Clearly this slows the whole thing down, but then virtual addresses are a very nice feature. You can hide the pages used by other tasks or map the same memory to several tasks, making it shared memory.

These inclusion of these features implies that all resources can be assigned equally to each task. As there are problems with this in the 6502 (think of the stack), another concept should at least be mentioned. The IBM `Virtual Machine' (VM/*) series of operating systems emulates the entire computer's hardware resources for a single task (i.e. a task doesn't talk to the system via system calls, but by writing data into some I/O registers). These register accesses are trapped and appropriate action is taken. This means that the task can behave as if it owns the entire machine. This also means it must load its own OS to handle disk and other I/O (the second part of the "VM/*" naming scheme).

The Commodore PET and its successors, the VIC, C64 and 128, already contain some functionality of a "real" OS. On these machines, a single interface allows uniform file access across different devices (tape, disk, console). All of them are accessed via the standard OPEN / CKOUT / CHKIN / CLOSE system calls. However, I/O comprises only one part of an OS, as defined above. The Commodore 8 bit computers are single CPU, singletasking systems (for exceptions see below). Therefore, no process management is necessary. In addition, there is no memory management. All memory is assigned to the single running process. (Although sometimes the need for multiple $cXXX pages seems pressing.) The filesystem, an important part of an OS, is put into the floppy drive on Commodore 8-bit computers and is accessed via standard I/O over the IEEE bus.

One interesting exception is the old (IEEE488) Commodore disk drives. These drives have not one but two processors: one 6502 and a 6504 that run in parallel and share some memory. The 6504 is used as a floppy drive

controller that handles the low level disk I/O. The 6502 gets the commands from the bus and processes the `filesystem' task. By writing low level commands to certain memory locations, it sends commands to the floppy drive controller (the 6504) that in turn reads and writes the disk blocks. If you look at the 1541, for example, you can see that this concept still holds true. However, in the 1541, the interrupt routine takes the role of the drive controller. Ironically, this reduction in CPUs was done to save 1541. In its effort to cut costs, Commodore forced the single CPU of the 1541 to multitask, creating a bare operating system to support drive operation.

@(A): Modern Kernel Design

Early operating systems started with a monolithic approach. i.e. all the system functions were provided with one big binary. Modern UNIX systems- even Linux, which is not derived from the original UNIX source- use this concept.

A modern kernel instead has a microkernel design. A microkernel only provides the means of communication between different processes, not doing much itself. Some implementations even have the scheduler (!) or memory manager (!) running as a separate task. The kernel calls these processes to find out about free memory pages and which task to start next. This reduces the size of the kernel and allows greater flexibility. On the downside, the microkernel designs forces more messages to be transferred, slowing down operation somewhat.

One `famous' microkernel implementation is the current Mach microkernel. This kernel, and its derivatives, has been ported to many platforms. The PowerPC Platform OS/2 is based on a mach derived microkernel, as well as Linux for PowerPC Macintosh (mklinux). But, these are relatively simple ports of already existing operating systems. These mach `single servers' don't allow alternate OS system to run alongside or instead of themselves. On the other hand, the GNU Hurd operating system exploits the mach design to allow any server to be replaced by another.

@(A): The OS/A65 Operating System

Now let's get from the theory to practice...

@(A): The Kernel Implementation

When it comes to hardware design, the 6502 has a big advantage: It is a very simple CPU. With only a few support ICs, it is possible to build a fully functional computer (neglecting video and sound capabilities). On the other hand, the simplicity of the CPU has drawbacks. The 6502 has only three multi-purpose registers, and all are 8 bits. As such, none can hold a complete 16 bit 6502 memory location. Even the stack pointer is 8 bits, restricting the stack to the 256 bytes from $0100 to $01ff. The stack size and the absolute addresses are a severe limitation if you intend to develop a multitasking OS on this machine.

Because I was developing a new system, I could do anything I wanted to get around this problem. I solved the stack problem by using an MMU, a Memory Management Unit. (Although the used chip, the 74ls610 is stated to be a `Memory Mapper' for paged memory mapping, I call it a `Memory Management Unit'...). The upper 4 address bits are used to select one of 16 8-bit registers. (The 74ls610 has 12-bit registers, but only 8 bits are used, for obvious reasons.) The output of the registers were then used as the upper 8 address bits, extending the total accessible memory to 1 MByte. The CPU could switch each 4 kByte page to any of the 256 pages available by changing the register values in the MMU. Oops - just introduced virtual addresses to the 6502 ;-)

For each task, new memory is allocated and saved in the task's page table. When a task is activated, the MMU registers are loaded with these values, giving each task its own memory environment. In the described OS, the memory `manager' is part of the kernel, although a quite independent part. The virtual addresses in the opcodes are translated to the real addresses through the contents of the MMU registers.

The tasks are handled by the environment routines. These routines set up the environment tables used by the scheduler. The (round robin) scheduler performs the task switching and decides which task to run next. Preemptive multitasking is achieved by using the interrupt to switch between different tasks. The most important routines are the two kernel entry and exit routines. These sub-routines have to switch the pages and the stack pointer as well as preserve all other register values.

The tasks providing filesystem services register with the filesystem manager. They are then assigned drive numbers. Although UNIX filesystems

are virtual, where a user can reconfigure the system at any time, developing such a system for the 6502 would overly complicate matters. Different filesystems can then be used at the same time with different drive numbers. The drive numbers are translated by the filesystem manager when passing the message through to the filesystem task. Currently `fsiec' for IEEE488 (parallel IEC-bus) interfaced CBM disk drives, `fsibm' (for PC style disks) and `fsdev' for using devices as files are provided.

The interface to the hardware is provided by the devices. Devices are simply stripped off tasks and are called as subroutines only. A device-filesystem (`fsdev') task translates filesystem requests to the device interface, so that any device can be used like a file. The general structure can be seen in Fig.1.

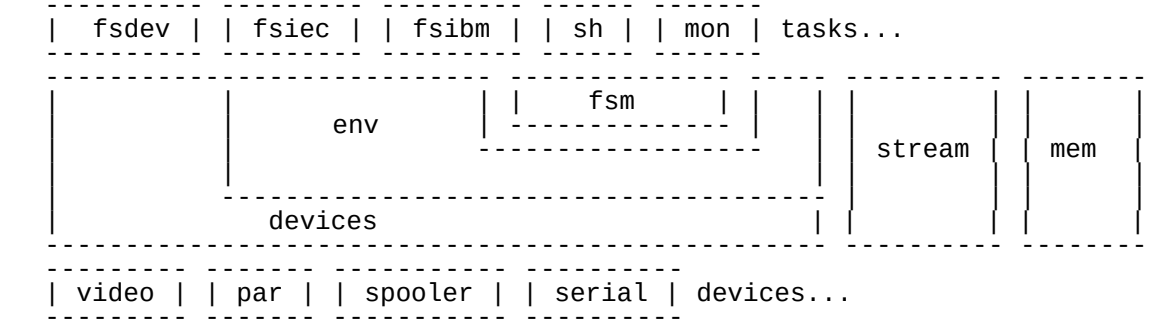


Fig.1: General OS structure. The devices and tasks make up the features of the system, while the kernel provides communications.
(fsm = filesystem manager, env = environment handling, task switcher)

In addition to executing code within the task, tasks also need to execute to communicate with other tasks or components of the OS. To communicate between tasks, a send/receive interface is provided. Using a rendezvous technique (the sender blocks till the message can directly be copied to the receiver and vice versa) the mechanism is kept simple, as no buffering is involved. Semaphores can be used for synchronization between different tasks. Data streams are used to pass data between tasks, and even between tasks and devices. Each task has a standard input, output, and error streams opened upon creation, analogous to the stream in UNIX systems. The shell can even redirect or pipe the output.

@(A): Program examples

The shell is a good example to show some of the capabilities of the system. As already mentioned, each task has three specially assigned streams. Filesystem tasks don't use them (and have them set to an ignored stream), but shells normally get started with these streams connected to a terminal device or a serial line device. The streams are normally opened by the task that `forks' the new task. On boot, the ROM contains some hints about which device number to open for a program. When a new task is started with a shell command, the shell has to open the devices. Normally the standard input and output streams used by the shell itself are registered for the new task. However, if given on the command line, other files can be opened and the streams for these files used as stdio streams.

When a file has to be opened, an OPEN message is sent to the filesystem manager. This part of the kernel translates the drive number and forwards the message to the filesystem task. The filesystem then tries to open the file and sends a reply message. The originating task provides a stream number with its first message. If the filesystem task succeeds in opening the file, it uses the provided stream to read or write the data to. If the file ends, the writing task closes the stream, which is recognized by the other end when there's nothing more to read. This works for read only and write only opens, but not for read/write opens.

@(A): Problems

Bootstrapping was the first major problem. How do you start a new computer and debug its OS if don't have an OS on the computer? From earlier systems I already had a small monitor program - directly burned into an EPROM - able to load binaries through a serial line. Getting the MMU (74ls610) was the second problem, because it was on the CoCom list, and it was not allowed to export to eastern countries. (Although I don't live in an eastern country, this posed some difficulties...)

After defining the necessary interfaces between kernel and tasks and kernel and devices, the design was quite straightforward, actually. One problem was the small number of registers in the 6502. For some of the kernel routines, as well as for the send/receive interface it was necessary to define a special buffer. This buffer is at an absolute address at $02XX, which is the same for each task. For systems with an MMU, this is not a problem after all. But it showed out to be a significant problem when porting the OS to systems without MMU, like the C64 (see below).

@(A): Operation without an MMU

After the system worked well with an MMU, I decided to build a stripped down version for systems without an MMU to better fit some `embedded applications' I had in mind. The system without an MMU is much more a multithreading than a multitasking system. Threads, as opposed to tasks, share the same memory, thus being able to change variables and data of other threads. But, on the other hand, two identical programs cannot run at the same time as with an MMU, unless they know they will together ahead of time.

The problem lies within the limited stack size of the 6502. Without an MMU, it is not possible to remap memory pages, especially the page with the stack in it. So the stack is divided into several parts, limiting the stack size of each thread, of course. Another problem is global, absolute addresses - like the send/receive buffer for example. As it would be too much of a rewrite and memory wastage to give each thread its own buffer, the send/receive buffer is now protected by a semaphore. A sempahore is a construct that allows exactly one thread to be in a certain routine or manipulate the protected data at a time. Semaphores originate from the railways, where it is important not to have two trains on the same rail, running in opposite directions...

@(A): Port to the C64

In addition to lacking an MMU, the Commodore 64 posed other porting problems. Only small changes had to be made to the kernel. The C64 kernel required an interrupt source for task switching. The video device had to be changed to support the C64 keyboard map and video interface. The hardware cursor used in my homebrew computer was replaced by a software cursor. The IEEE488 filesystem was first ported to the IEEE488 interface for the C64 and then to the C64 serial port. When stress testing the system I realized that I still hadn't ported the STDIO library - a few low level subroutines that make life easier. The library was mapped to most tasks and was called from the task environment, not from inside the kernel. Unfortunately, it used global variables - which broke the library when running on a multithreaded system without an MMU. Therefore, some routines have been changed, while others can only be protected by a semaphore.

@(A): Port to the C128?

Well, the C128 has more memory and even the capability of remapping the stack and zero page to other locations. In a simple expansion of the C64 version, this could be a way to raise the limited stack size to the full possible 256 bytes. Then, other ideas come to mind. The original memory management is made for a system with MMU and is quite useless without an MMU. What is missing is a call to get a contiguous memory block of more than a memory page in size. Then such a large block could be allocated for a new task to load the binary. The binary itself must then be relocated to fit the new address range. Unfortunately, plans to extend the system calls or add relocation capabilities do not exist at this time.

@(A): Conclusion

The OS/A65 operating system provides multitasking and multithreading capabilities with a modern kernel design for a 6502 CPU. The OS can be used from embedded applications to desktop systems. A shell provides modern I/O redirection and piping capabilities. Filesystems for Commodore disk drives and PC-style floppies are available. For me, it was a real adventure to design a completely new computer and operating system the way I wanted them designed. I also learned a lot about operating system design - maybe you have learned a bit as well. If you are interested in it, more information is available at:

http://www.tu-chemnitz.de/~fachat.

==========

@(#)usenet: UseNuggets

COMP.SYS.CBM: The breeding ground of programmers and users alike. Let's see what topics are showing up this month:

@(A): Let's Poll Together

Throughout the past few months, Paul Allen Panks has been conducting a poll on Commodore Business Machines' greatest success stories and most momentous flops. Although some biased opinions exist, many have agreed that the C64 was a success, while the 264 series (Plus/4 and C16) was a flop. After that, however, and few agree.

@(A): Ymodem vs. FX, Round -1

The many people who use Craig Bruce's ACE environment know that he recently added support for a special transfer protocol, FX. Proprietary in nature, FX supports very large buffer sizes and can achieve throughput of 200% or more over standard protocols like Ymodem or Xmodem. The downside of FX is the necessity of compiling an FX "server" on a UNIX host in order to utilize the protocol.

While not newsworthy in itself, a discussion about which standard protocols are fastest kicked up some dust. Many were inquiring about DesTerm support for Zmodem, causing Ismael Cordeiro to note that the DesTerm protocol implementors chose to optimize existing protocols rather than introduce new ones.

A lively debate started, as Craig Bruce noted that even the fastest implementations of Ymodem were no match for FX. Ismael countered by calling the comparison unfair. Ismael noted the drawbacks of FX being proprietary and not available for all Commodore users.

Also, Ismael explained the reasons for FX's increase in throughput over standard protocols. Packet size was a large factor, as FX uses a much larger buffer size. However, FX suffers when retransmissions are necessary, since the time between handshakes (which occur between packets) is much longer. When using a comparable packet size, FX and Ymodem are competitive.

@(A): Operating System Support

In last issue's USENuggets, we discussed the conversations stemming from the proliferation of operating system ideas on comp.sys.cbm. (C=H#12, Reference: usenet) We noted that many expressed a need for programmers to support the ACE computing environment, written by Craig Bruce. Upon noticing this, Craig responded:

> "I, of course, support the idea of other people building more applications for the ACE environment. I also support the idea of using ACE applications with other operating systems. ACE was built on the idea of providing a well-defined Application-Program Interface (API), and any alternative OS that can emulate the ACE interface (using a "middle-ware" layer of software) can run all of the existing ACE applications. Thus, a new operating system can have a base of (a few) high-quality programs available instantly (high-enough quality that even _I_ use them). Admittedly, I have to update the documentation on the ACE API, since it changed in Release #15, but the basic functionality will always be the same.
>
> In addition, I also support the idea of other people using ACE code inside of their own operating systems. Why re-invent the wheel? Especially useful may be the dynamic-memory stuff and some device drivers. ACE is Public Domain software, so you can do with it whatever you please."

@(A): The "More Power" Swiftlink (An Update)

As well, Craig followed up to our story last issue on the "hacked" Swiftlink that could do 115,200 bps. (C=H #12, Reference: usenet) Craig noted that ACE #15 supports the modified Swiftlink and that the code in ACE handles the new speeds "flawlessly".

@(A): And Speaking of Operating Systems...

Since the last issue of Commodore Hacking, at least two more operating systems have been announced. One, OS/A65, is detailed in this issue of Commodore Hacking (Reference: os). Another, called COMMIX 2, will encompass an object oriented operating system. The system is comprised of multiple sub parts, including:

Networked X Input/Output (nXIO), the communications sub system

COMMIX Object Format (CXOF), an object and code description format
nXIOtee, the object oriented programming language.

For more information on this networked OS design, check out its WWW site
at: http://www.cynapses.com/ry/cx2/cx2home.html

==========

@(#)uqwk: Using UQWK with QWKRR128
by Gaelyne R. Moranec (gaelyne@cris.com)

@(A): Introduction

One of my first priorities when joining an Internet service was to find a way to utilize the QWKRR128 offline mail and news reader to read Internet email and USENET newsgroups. Like all QWK offline readers, QWKRR128 is commonly used with Bulletin Board Systems (BBS). A user dials into a BBS, selects which groups and what email to download. The BBS program then gathers and compresses the user's requested messages into a file called a QWK packet. The user downloads the resulting packet, and then runs QWKR128 or some other QWK reader on the packet. Thus, users can read email and news offline and reduce connect time. Replies are also handled in much the same way, allowing the user to read and reply to messages without tying up the phone.

What happens when we replace the BBS with the Internet? Well, for a while, making the switch meant shelving QWK offline readers. However, as with all problems that occur on the Internet, this deficiency was soon remedied by Steve Belzack, who wrote the Unix QWK system, called UQWK. It allows Internet users to package up Internet email and USENET newsgroups into QWK packets for use with QWK readers like QWKR128. Like its BBS counterpart, UQWK also handles reply packets from the QWK reader.

@(A): Finding UQWK

You can find out if your system already has UQWK by typing any of the following - if one command doesn't work try the next one.

```
where uqwk
whereis uqwk
which uqwk
find uqwk
```

If your system has UQWK installed, DON'T run the program until after you've read the manual for it. UQWK requires command line switches to work and defaults to emptying your mail box, which isn't nice. To read the manual, type:

```
man uqwk
```

It's a good idea to create a text file in your home directory with the manual so you can download, print, and review it offline. The command to do this is:

```
man uqwk >> uqwk.manual
```

Then, to read it you type:

```
more uqwk.manual
```

To download it with Ymodem, the command is:

```
sb uqwk.manual
```

If your system doesn't already have UQWK available, you may be able to get the file and compile it for your personal use. Because there are so many versions of Unix to deal with, I cannot help you with compiling it for use on your system. If in doubt, give the file to your system administrator and ask him or her to install it.

The FTP site is: gte.com
Directory: /pub/uqwk/uqwk1.8.tar.Z

Be sure to get both UQWK and the README file. The text file will tell you step by step how to set it up on your account.

@(A): Using UQWK

I use two Unix script files when I use UQWK, named "getmail.script" and "sendmail.script". I keep these text files in my home directory. I

had to change the permissions on them so Unix would see them as "executable" files. The command for this is:

```
chmod +x filename
```

or

```
chmod 700 filename
```

You will need to make changes in the files so that they represent the BBSID used on your system. For instance, CRISINET is the BBSID on my system and is used in the examples below.

When you use the getmail.script the first time, just use an arbitrary name for the name of the .qwk packet, but change your script after you know the correct BBSID to use. Be sure to use proper upper or lower case *exactly* as it appears in your control.dat file for any references to your .REP and .msg files. This may not always work, however, as it depends on your terminal program. Some CBM term programs will maintain the same casing as is used by PETSCII, while others will convert them to ASCII. If yours changes the filename, be sure to change the appropriate lines in your script files so UQWK and other utilities can find it.

@(A): Scripts To Get You Started

```
# ----------------
# getmail.script
#

rm crisinet.qwk
uqwk +r +m +n +e
arc a crisinet.qwk *.dat *.ndx
sb crisinet.qwk
rm messages.dat *.ndx

# ----------------
```

Notes:

`rm crisinet.qwk` - This removes any previously created .qwk packet. it is in lower case, as since we name this file ourselves, there's no need to make it uppercase.

`uqwk +r +m +n +e` - The command to tell UQWK what you want it to do.

+r keeps UQWK from deleting your Email and marking your newsgroup messages as read.
+m process Email.
+n process newsgroups
+e tells it to create a control.dat file listing ONLY those subscribed newsgroups.

* Also you can use -m or -n so UQWK won't process
* mail or newsgroups. UQWK defaults to doing
* Email, but not newsgroups. (+m and -not)
* The +e switch is a must for QWKRR users, as
* this list gets loaded into memory and reduces
* the amount available for reading messages.

`arc a crisinet.qwk *.dat *.ndx` - This creates an ARC archive of the files UQWK has created. QWKRR users don't need to include the *.ndx files, but it's included here for those who use other offline mail readers. ... Heathens! :-)

As mentioned previously, although the BBSID is "CRISINET", since we are creating the archived file, we can leave it in lower case for our own convenience.

`sb crisinet.qwk` - This begins a Ymodem download of your QWK packet. You have to start the transfer with your terminal program manually.

`rm messages.dat *.ndx` - This removes the messages.dat and *.ndx files from your directory. If you have sensitive Email you don't wish others to view, this

prevents anyone from reading it.

```
# ----------------
# sendmail.script
#

rb
unzip CRISINET.rep
uqwk -m -n -Rcrisinet.msg
rm CRISINET.rep

# ----------------
```

Notes:

rb - This begins a Ymodem upload so you can upload your Reply packet. You have to start the upload with your term program manually.

unzip CRISINET.rep - If you've <Z>ipped your reply packet, this is the command to unzip it. When QWKRR creates the file, it honours the case of the BBSID, so the filename is in upper case.

uqwk -m -n -Rcrisinet.msg - This is UQWK command to process a reply packet. The -m and -m switches tell it NOT to process your Email or newsgroups into a new batch of mail to download. This file (crisinet.msg) is within the "REP" packet. It is lower case.

rm CRISINET.rep - This deletes the .rep file from your directory. UQWK automatically deletes the *.msg file.

You can also create these scripts with your term program. Either way works.

When you review the UQWK manual, you'll see the commands and should be able to follow the script file and make adjustments to suit your needs. You can have UQWK create QWK packets for Email, newsgroups, or both. Also, you can have one script file that sends your replies then creates the next batch of QWK mail for you.

@(A): Safeguarding Your Email

On one system I use UQWK with, I can back up my Email file, something I recommend especially when you first start using the program. To back up my mail file, I copy the mail spool file to a local temp directory. The actual path string for this varies depending on the type of Unix system you are using. For me, this works:

```
cp /var/mail/username ~/temp/filename
```

On another system, I can't make a backup of my Email file, as the system doesn't allow users to move or copy mail files. However, I can use a command for UQWK that tells it not to erase my mail or newsgroup articles. If you use the read-mode only command, you have to delete Email manually, and mark newsgroup articles as read.

NOTE WELL:

UQWK uses your .newsrc file to find what groups you are subscribed to. ALWAYS upload and process your current Replies before subscribing or unsubscribing to newsgroups, or else you will have your replies going to the wrong newsgroups.

@(A): The Files UQWK Creates

UQWK only creates the base QWK mail files, which are "control.dat", "messages.dat" and files that end with "*.ndx" (*.ndx files are not needed for use with QWKRR). If you want to you can archive the files QWKRR needs, or you can download the *.dat files uncompressed. The getmail script file covers creating the arc file and beginning a Ymodem download.

I compress my mail using arc, as I have a program that will automatically dissolve my QWK mail and start QWKRR. The program is called QPE, and can be found in the archive NZP12817.SFX. If you arc your mail packet, you will need an ML program found in the archive CSX01.SDA. I could use Zip, but my ISP's Zip program creates only PKZip

2.04g files, and Commodore users don't yet have a program that will unzip these.

@(A): Replying To Email

By default, QWKRR doesn't display any data after an "@" symbol in the headers. To be able to see the complete Email addresses (a must for Internet use), first load but don't run QWKRR. Type:

```
poke 49169,255
```

Then save the program using a different name (such as qwkrrinet), just in case you've made an error when entering the values.

@(A): Long Email Addresses

If the Email address of the recipient doesn't fit in the "To:" field, you must use other addressing methods. Erase the name in QWKRR's header and substitute the person's first and last names, or any two words with a space between them. Do NOT have a "." or "@" here if the full Email address is too long to fit in the field. If you do, UQWK assumes it's a valid Email address. The reason you want two words instead of one is so the program doesn't assume you're sending local mail on your ISP.

On the first line of the message, type:

To: user.name@anywhere.com

Begin your message on the following line.

Hint: Type "To: " on the first line. Quote enough of the message so the Email address is on the screen, and then move the address so it is in place after the "To: ".

There is a space between the colon and the Email address.

@(A): Sending Newsgroup Articles

The only thing different from Email you'll need to do is make sure that your articles have the word "all" or "ALL" in QWKRR's "To:" field.

Messages from almost any QWK offline mail reader do not conform to Internet standards for newsgroup articles, as QWK was originally designed for Fidonet only. You can still post articles with these programs using the above method of placing "all" in the "To:" field.

For those who want their articles to conform to the Internet specs, you can have UQWK look to the body of your message for the header information by using the +X switch. This will let threaded newsreaders properly add the article into an existing thread. This is only for those who are well experienced with RFC-1036, the "Standard for Interchange of USENET Messages" and RFC-822, the standard for Internet Text Messages. These documents can be found on the web at: http://www.internic.net/rfc. In the future, I'll be adding information to QWKRR's web site on how to create articles that do conform to this standard.

QWKRR has a known bug when it comes to quoting lines that are over 255 characters long. This bug often appears when replying to newsgroup articles, as the "Path:" line often exceeds this. The next version of QWKRR will not have this problem. To reply to a newsgroup article that has a long pathline, export the article as a temporary text file, then import it into the message. e<X>port is a function only available to registered QWKRR users.

@(A): A known UQWK Quirk for QWKRR users

When importing text that has a "message" header on it (i.e., all the To, From, Subject etc.), UQWK makes the assumption that a new message has started. To avoid having your message split at this stage, indent the To/From info in the imported text about 4 columns.

@(A): Sending Your Replies

Most Unix systems can unzip reply packets that have been Zipped by QWKRR. It can also handle files that are ARC'ed if you use the QPA program. UQWK doesn't require this. All UQWK knows about is the *.msg file within the .REP file. It is possible to choose <L>ink within QWKRR and upload the resulting *.msg file, BUT if you do this, you may have problems with Xmodem padding (also Ymodem) added to the end of the file by your term program. This extra padding will cause you to receive an

Email bounce as UQWK tries to interpret the padding as a message. It's easier to <Z>ip the replies then let your script file unzip them.

@(A): UQWK and Signatures

When posting articles to newsgroups, UQWK will append your .signature, but if it doesn't like the length of your signature, it will not post the article. (I don't know the length it will accept). You may want to change the filename from .signature to .sig and use a QWKRR macro for your signature instead. (Be sure to change your settings for other programs like Pine so it will look for a file called .sig, though).

@(A): UQWK and Newsgroup Subjects

There is a UQWK version that doesn't accept newsgroup articles created with QWKRR and complains that the subject line is incomplete or incorrect. So far the only cure I've found is to use an older version of UQWK that my system has online. UQWK version 1.8 does not have this problem, and after checking FTP sites, it appears my current ISP is using a customized version. If I find others have similar problems and find a cure, I'll post info regarding it on QWKRR's WWW site. http://www.msen.com/~brain/guest/Gaelyne_Moranec/qtoc.html

@(A): Conclusion

While reading BBS news and email offline is a blessing, it is almost a necessity on the Internet, where the level of email and news can be overwhelming to the online reader. UQWK and QWKR128 make a powerful combinations that help you manage your time effectively yet still enjoy the pleasures of keeping current on all the Internet has to offer.

==========

@(#)fido: FIDO's Nuggets
by Geoff Sullivan (geoff.sullivan@tbbs.bcs.org)

The CBM GEOS, CBM, and CBM-128 FIDONet echoes are places where Commodore users unite. Let's see what they discussed over the past few months:

@(A): GEOCable and Printers

GeoCable was a product originally marketed by Berkely Softworks to eliminate the need for a serial interface for non-Commodore printers in the Geos environment. It also speeds data transfer from computer to printer. Well, some users decided to test this speed increase and found that what was accepted before may not be true in all cases. Many scientific, and not so scientific, test results showed that the speed of printing may have more to do with the type of data being printed and the buffer size of the printer, than with the actual method used to get the data to the printer.

Lately more programs outside the Geos operating system are sporting printer drivers that support the GeoCable. As Phil Heberer aptly puts it:

> "Most of us GEOS users know the obvious benefit of using a geocable when printing from GEOS, but I'm also happy to see many programmers adding gc support to their programs. I can now use my geocable with nearly ALL of my favorite CBM programs that I currently use besides GEOS (i.e. Superscript/Superbase, TWS128, FGM, BROWSER and ACE15) If Maurice Randall gets 'The Wave' finished for GEOS, it will round out my applications quite nicely!"

Many users are building their own cables as well. Some users are discussing the need for drivers that will work with the Hewlett-Packard PCL language that is becoming more prevalent now that Commodore users are fooling around with ink-jet and laser printers.

@(A): DESTERM

Matt Desmond has recently posted a message on FIDONet confirming his work on a version 3.0 of Desterm. He has also stated again that it will have hardware flow control and enhanced REU support. It will NOT support any transfer protocol beginning with the letter Z.

@(A): EZ Loader v3.2 Released

David Schmoll announced the release of an upgraded version of his EZ Loader program for the 64 or 128. It is designed to help you access your most used programs on any disk or fixed drive through a single menu. Although most useful for CMD drive owners, it can be used with any

Commodore drive. It has too many slick features to be mentioned here but certain ones are disabled on the downloadable version. They can be activated by registering the program. It is available via FTP from ccnga.uwaterloo.ca in /pub/cbm/util128/ and possibly on local BBS's by now.

@(A): Alternate Character Set Access

One user was toying with the idea of storing multiple character sets in the VDC 64K memory of his C128 and swapping between them by simply changing the register address. His aim is to perfect this for display applications for various programs such as character set editors. Rod Gasson suggested an alternate scheme would be to swap the entire stored character set from the VDC ram into the default page at $2000. He says that the VDC's block move is very quick and it allows mixing of characters from more than one set.

@(A): Internet

Some folks have reported problems downloading binary files via Lynx or FTP through UNIX servers with their Commodores. Ismael Cordeiro had some suggestions for these MIME type problems. For those with shell access on a UNIX system he suggested using FTP with a customized MIME type file:

> "...create a text file named '.mime.types' in your home directory with one line:
>
> application/octet-stream sfx sda arc prg cvt lnx
>
> If you don't have shell access and Lynx is the user interface...the only thing to do is ask the system administrator to include the above line in the system's mime.type file."

@(A): Miscellaneous

Among the miscellaneous topics being discussed on FIDONet is the use of a C64/128 for ham radio communications. This is a rather popular use for the 64. The program being discussed is Digicom. Many newcomers are still asking questions of the "old timers" concerning Desterm setup with high speed modems, REU expansion, and off-line mail reading and replying. For a "dead" machine, it is surprising to see how many are being dragged out of closets, dusted off, and booted up!

So, that's a glimpse into the world of FIDO, the wonder dog of networks, for this time.

Here, boy....

==========

@(#)pal: Brad Templeton: The Programmer's Friend
by Jim Lawless (cjbr@gonix.gonix.com)

The following text is an interview held via e-mail with former C64 software author Brad Templeton. Mr. Templeton is the author of the PAL assembler and the Power productivity tool.

Mr. Templeton is the founder and current CEO of ClariNet, a networked newspaper with over a million subscribers. Please refer to the references at the end of this text for Internet resources detailing his accomplishments.

Q: Were PAL, Power, and C Power fruits of your imagination, or were you contracted by Pro-line to write them?

A: C Power was a C compiler written by Brian Hilchie, nothing to do with me.

> But POWER and PAL (Can't recall which I did first, probably PAL, but POWER was the one sold first.) were done on my own. Professional Software licensed Power for the Pet and Pro-Line licensed it and Pal for the C-64.
>
> Actually, I think I wrote a quick cross assembler in B (the predecessor language to C) to run on the mainframe at my university first, and wrote the early version of PAL in that. Then of course moved it to the Pet so that PAL could assemble itself -- always the big moment in any language development. My memory is getting dim, I might have started from an Apple based assembler. I know I wrote a cross assembling, one-pass version of Pal, with macros for Unix a few years later but just used it to develop stuff for the C64.

(Most people are startled to learn that C compilers, even the very first one, are usually written in C, and so on. You bootstrap by writing a very simple one using an existing tool, then get it going and then enhance it.)

Q: PAL was/is one of the most widely used assemblers for the C64 (and I assume the PET). Had you written any assemblers before PAL, or did you just happen to create a darn good product "coming out of the starting gate"?

A: No, I hadn't written any assemblers other than the cross assembler. Before that however, I had developed Time Trek, a game for the Pet, Checker King (a game) for the Atari 800, Apple][and Pet and the Atari 800 graphics for Microchess.

Q: In the days of PAL and Power, were you actually making a living writing software for CBM machines or was it sort of a part-time excursion?

A: Well, I was a student at the time. But after graduating, it was enough of a living to be able to work on other projects, and eventually get the contract to develop my next product, Alice Pascal, in 84.

Q: What were some of the biggest problems marketing your CBM software? (Was piracy an issue?)

A: Piracy was somewhat of an issue. The big mistake with Power was doing demos at some pet user groups before I was ready to sell it. Bill Seiler of Commodore saw a demo I did at the silicon valley PUG, and added some of the best features to Basic-AID, which Commodore gave out for free. Power was better than Basic AID but a good free competitor didn't help.

It was still a hobbyist market, not nearly as big as the computer industry grew to be.

Q: When and why did you finally abandon development efforts geared toward the C64?

A: The machines faded away and the IBM based machines clearly took the lead for more serious applications. If you wanted to do things that took more than a few kilobytes, or work in C, the C64 wasn't really an alternative.

I did some games for the C64 but never went anywhere with them.

Q: With C64's showing up at garage sales and emulators available on a wide variety of machines, a renewed interest in that little machine is experiencing a rebirth. Do you have anything you'd like to say to a new generation of C64 hackers out there?

A: On one hand I am shocked, since vastly more powerful computers are of course available very cheap, garage sales or otherwise. However, there was a certain excitement to a small computer that one person could fully understand and work with like the Pet or C-64. If you view the computer as a hobby or a toy, it doesn't have to be the most advanced thing, what matters is that you have fun with it.

I certainly wouldn't advocate Windows programming to the ordinary start-up hobbyist but such people can have fun on a C64.

For more information on Mr. Templeton's current endeavors, the following WWW documents may be of interest to you:

An Interview with Brad Templeton
URL: http://info.acm.org/crossroads/xrds2-3/templeton.html

Brad Templeton's Homepage
URL: http://www.clari.net/brad/

==========

@(#)surf: Hack Surfing

For those who can access that great expanse of area called the World Wide Web, here are some new places to visit that are of interest to the Commodore community. In early 1994, when the US Commodore WWW Site started, the number of sites online that catered to Commodore numbered

in the 10's. Now, the number is in the 100's. What a change.

If you know of a site that is not listed here, please feel free to send it to the magazine. The following links have been gleaned from those recently changed or added to the US Commodore WWW Links Site (http://www.msen.com/~brain/cbmlinks/).

To encourage these sites to strive to continually enhance their creations, and because we like to gripe :-), we'll point out improvements that could be made at each site.

@(A): Companies

o The Official DesTerm 128 Page
   URL: http://www.ionline.net/~mdesmond/desterm.html
   Here is where you will find the latest scoop on the popular terminal emulation program for the 128, as well as information on the newest release, Desterm 128 3.0. As well, you can download Desterm 2.00. C=Hacking gripe: There isn't much information on the 3.0 version.

o Keyboard Studio
   URL: http://www.cu-online.com/~gwilson/
   Gordon Wilson's company motto is: "Large enough to get it done; small enough to care." That sits well with us. This small site announces Mr. Wilson's Commodore repair facility to the world. It offers basic information about the type of repairs possible and what other services are offered. C=H gripe: We wish there was more detailed information on repair services, like pricing information.

o Novaterm 9.6
   URL: http://www.eskimo.com/~voyager/novaterm.html
   For a guy who just released a new version of his popular C64 terminal emulation program, Nick Rossi has managed to put some effort into this site. The site is flashy, but can be viewed with text browsers as well. The information here includes a rundown on Novaterm 9.6 features, details on who helped write it and how to purchase it, and links to obtain the 9.5 release. Of special mention is the fully indexed HTML online documentation. C=H gripe: For those who want to order with a credit card, the site refers to a list of authorized Commodore dealers that we couldn't find.

o Omni 128 BBS Software Home Page
   URL: http://www.nwlink.com/~bbell19/omni128.html
   At this site, Brian Bell presents an overview of his Bulletin Board System Software and updates on releases. Additionally, information on capabilities like "Echo Net" are present. C=H gripe: We couldn't find out how to purchase the software or how much it costs.

@(A): Publications

o DisC=overy Home Page
   URL: http://www.eskimo.com/~drray/discovery.html
   We'll save a review of the magazine for "Hacking the Mags" (Reference: mags), but the publication does tout its own WWW site. It's pretty bare at present, but it does have links to both a text and also a compressed version of the Premiere Issue. C=H gripe: We didn't expect much here, but we do hope the publication offers an index or list of articles here at some point.

o 64'er Online
   URL: http://www.magnamedia.de/64er/
   This site presents information about the German Commodore publication. The layout is nicely done. The July issue is currently featured, with information on the contents and an index of articles. Alas, the site is for German readers only, but we expected no more. For those who can read German, ordering information and pointers to other products are available. C=H gripe: The site leans a bit heavily on graphics, making it slow to load.

o Commodore Online Information Network (COIN!)
   URL: http://people.delphi.com/cynrcr/ccs.html
   This site offers information on the COIN! disk magazine. Information on the magazine is presented, and links to the 2 most recent issues are provided for your downloading pleasure. A small description is given detailing the contents of older issues as well. C=H gripe: White text on a black background takes a bit of time to get used to. However, text mode users won't notice :-)

@(A): Demo Groups

o Millenium Home Page

URL: http://marie.az.com/~waveform/millenium.html
This site shows off screen shots of the demo groups' creations. The site is nicely done, with many screen shots and nice graphics. C=H gripe: How do we download the demos?

o Demo/Revenge Distribution Site
URL: http://hack.lakeheadu.ca/~revenge/index.html
Demo groups tend to provide the splashiest sites, and this one is no exception. The graphics are nicely done, but the content is available to all text-mode browsers as well. Links to demos are provided, as are links to other sites of interest. C=H gripe: With limited time to download, could we get a small description of each demo to help us pick?

@(A): Reference Works

o The C64 Games WWW Home Page
URL: http://www.student.nada.kth.se/~d93-alo/c64/
Screen shots are provided for a couple of C64 games, and clicking on the names reveals detailed information on the games and its gameplay. Music from many C64 games is present, as are tips and hints for playing some vintage Commodore games. C=H gripe: The name of the site is a bit misleading, since the list of games isn't that extensive.

o Poldi's Projects - LUnix
URL: http://rpool1.rus.uni-stuttgart.de/~etk10217/proj.html
UNIX on a 64. Don't even think that it cannot be done. Daniel Dallmann has already proved it CAN. This site details the entire project to execute a multitasking OS on a 64 from kernel to device driver. In addition, some of Daniel's other projects are detailed at this site. Daniel has developed a fast soft-80 screen driver for the C64, and the code with detailed information is available here. Schematics, code, comments, and an overview for Daniel's 9600 bps serial routines are available here. These routines have also been incorporated into Noavterm 9.6. Finally, Daniel has developed a basic implementation of the Serial Line Internet Protocol (SLIP) for the 64. Code and information are linked off the site. Many of the projects include screen shots and schematics. C=H gripe: A high level overview of some of the projects would help first time surfers.

o OS/A65 Computer and Operating System
URL: http://www.tu-chemnitz.de/~fachat/csa/
Andre Fachat's work on a multi-tasking OS and a home built 6502 based computer system are outlined at this site. The software is detailed in Andre's article elsewhere in this issue (Reference: os), as well as on the site. The full text is presented on the site, with an indexed overview. C=H gripe: We couldn't find a link to the bare 64 binaries.

o Technical SID documentation
URL: http://stud1.tuwien.ac.at/~e9426444/sidtech.html
For the SID-savvy of the bunch, this site offers a technical discussion of the 6581 SID IC and descriptions of the various waveforms with mathematical treatment. C=H gripe: The presentation is pretty basic.

o Commodore Product Source List Issue #5, On-line Edition
URL: http://www.televar.com/~rjlong/
Roger Long has placed his Commodore products SourceList Online. The online version, which is updated more frequently than the printed version, contains a wealth of information on where to find hardware, software, and supplies for the Commodore computer. C=H gripe: An alphabetical index would be nice.

o Carrier Detect
URL: http://www.swt.edu/~ez13942/bbs/cbmbbss.htm
For all the BBS sysops or ex-BBS sysops, this page will certainly bring back memories. A History of BBS in the 1980's is given, followed by an extensive review of various BBS systems. Each review includes statistics and screen shots. C=H gripe: The background makes the graphical version a bit rough on the eyes. As usual, though, text viewers won't care :-)

o Bacchus' List of 64 related PC and Amiga tools
URL: http://www.ludd.luth.se/~watchman/fairlight/c64/tools2.html
If you regularly use PC or Amiga platforms to develop Commodore executables, this site is for you. It gives a list of many PC and Amiga utilities to help the cross platform developer. Many programs are available, and they are all sorted into categories based on function. C=H gripe: We wish there were more detailed descriptions.

@(A): Individual Commodore Users

o QT's Dream Space
   URL: http://www.lm.com/~qt/
   QT is a demo lover, and it shows. There are links to demos, lists
   of new releases, links to demo magazines, and even a tribute to
   "Coder's World", a demo coding tutorial. In addition, there are PC
   versions of 64 compatible ZIP and LYNX compression programs to give to
   your computing challenged PC friends. C=H gripe: QT likes SunSoft's
   JAVA mascot, and has him displayed on the site. It's a bit misleading
   for those who expect JAVA information wherever the mascot is displayed.
   (He is cute, though).

o Don's and Mex's Game Page
   URL: http://blitzen.canberra.edu.au/~dryan/c64main.html
   In a page true to the Commodore, this page's headings are done with
   text screen shots from a C64. We are impressed. Lots of games are
   presented on this site, with basic information and screen shots
   provided. Links from each game allow the viewer to download the
   binary. C=H gripe: Some of the games are copyrighted and commercial.
   At the very least, a warning should be placed on the pages.

o Welcome to the World of Saz
   URL: http://www.wonderland.org/~sarah/
   Sarah Dalrymple has provided the WWW surfer with a plethora of
   information on the Commodore VIC-20. Pictures of units and peripherals
   are featured, as are some historical facts and links to other VIC-20
   sites. C=H gripe: The Games/Programs link wasn't functional.

o Triumph's Zone
   URL: http://www1.usa1.com/~triumph/
   This page show us how one person uses a Commodore system. As well,
   this page demonstrates the immense pull CBM machines have on users.
   Triumph had left for greener pastures when a friend "re-introduced" him
   to the Commodore.   For the adventurous, there are plans here for a
   C64 laptop computer under development. C=H gripe: The color
   scheme leaves a bit to be desired, but text browsers won't care.

@(A): Change of Address

o LOADSTAR has moved (AGAIN!) to http://www.loadstar.com/

o Marc-Jano Knopp's CBM WWW Site has changed (AGAIN!) to:
   http://www.student.informatik.th-darmstadt.de/~mjk/c64.html

o Richard Cunningham's Color 64 BBS Home Page has changed to:
   "Tim Allen's (Dynamite) Commodore Color Pages" and is now at:
   http://www.indirect.com/www/dynamite/color.htm

==========

@(#)demo: Dim4: A Mind Expanding Experience
          by Stephen L. Judd (sjudd@nwu.edu)

@(A): Introduction

"What in the world was I looking at? What the heck is your
code doing? How do I meet smart and ferociously gorgeous women like
you do?"

The last question I cannot answer, but this little writeup,
along with some pedantically well-documented code, can clear up the
first two, I think. This will not be a very dense writeup, honest!
Look to the code for more detail, and any equations below can be
skipped without problem.

In case you didn't know, dim4 was my entry into the recently
held 4k demo contest. For more info on the contest, as well as the
other entries (17 entries in all), seek ye the Driven home page at
http://soho.ios.com/%7ecoolhnd/

First, very briefly, the keypresses have the following actions:

   4   -- Turbo mode
   D   -- Normal mode (4D + 3D rotations, and nice and casual)
   F4 -- 4D-mode. All "3D" rotations are halted
   R/S -- 3D-mode. All "4D" rotations are halted
   .   -- Dotted line toggle
   Space advances to the next object.

The code is 4095 bytes long, and was a royal pain to get working after
compression. The music is Prelude #2 from The Well-Tempered Klavier by
J.S. Bach. I borrowed (and improved) the line drawing routine from the

cube3d programs and stole the patterns out of Polygonamy, otherwise the code is written from scratch, including the music routine. That crazy third object has fourteen sides in 3D, and the 4D object alone has 32 points with 96 lines connecting the points, so well over 100 lines are being drawn at a time. I was sorely tempted to put a docking bay on one of the sides of the 3D guy (a little "Elite" humor there) but ran out of time and room. After decompression the code expands a little bit, and in the end it leaves the 8k between $4000-$6000 free, but uses pretty much everything else.

The first object is a 4D cube, often called a hypercube. You can see a small cube inside of and connected to a larger cube. If you look a little closer, you may notice that in-between the two cubes are some more cubes. (When you slice a 3D cube, you get a 2D cube -- a square. When you take a slice of the hypercube, you get a 3D cube). As it rotates along its fourth coordinate, the cube folds in upon itself. One way to look at it is that the cubes start to change positions -- after 180 degrees of rotation the inside cube is on the outside and the outside cube is on the inside. The hypercube has literally turned inside-out.

The program works fine in PAL and NTSC, although PAL folks will get the tune playing at the wrong speed and transposed into a different key.

Oh yes, one thing I really like is the background on the second object-- on my 1084 it looks like rope. This is a consequence of the way VIC generates colors -- extra colors outside of the normal 16 are being generated, because two hires colors are being placed next to each other. If you look at it on a black and white monitor, you will just see thick diagonal lines. This very much surprised me when I first saw it! Find the March 1985 IEEE Spectrum article for more information on why VIC behaves this way.

Finally, you may notice some little glitches from time to time in drawing the 4D objects. That is my safety valve and keeps the program from literally destroying itself, in sometimes spectacular fashion. Oh well.

@(A): A Handy Glossary

Polygon: A rectilinear closed plane figure of any number of sides.

Vector: A directed line segment having magnitude and direction.

I do not know how the term "filled vector" came into vogue, but it is meaningless, not to mention a little silly -- what would an "unfilled vector" look like, two points with an arrow at one end? One may as well talk about filled lines and filled points.

Thus, I plead with the community to not refer to polygons as vectors and filled polygons as filled vectors. Polygons need your help, and have been discriminated against for too long now. Just one small donation on your part of a correct mathematical reference can help save the lives of one, ten, even hundreds of polygons, both abroad and here at home. Individuals wanting to contribute more may sponsor individual polygons; a kit will be sent to you containing the name of the polygon and at regular intervals a picture of the polygon will be sent to you, so you may monitor the progress of your particular polygon. Some polygons are created unclosed, and some do not get the necessary ink or programming skill to properly fill them, but be it a quadrilateral or decagon, trapezium or parallelogram, with your help we can eventually make all polygons closed and full, for a better, more civilized world. Thank you for your time, and God bless all the little geometrical constructions, no matter their dimension or configuration.

@(A): The Idea

This program displays a representation of some four-dimensional objects -- four 4D objects, as a matter of fact, each one of them a 4D analog of a three-dimensional object. Each screen contains four symmetry-related 3D objects and one 4D analog of the object, rotated and projected from 4D into 2D.

To describe the four-dimensional objects is not so tough. The 4D cube (the hypercube) is the first to be displayed, and it is the starting point for the later objects. It is also, I think, the easiest to see what is going on with. There is nothing really special about four dimensions -- with a 3D object each point is defined by three coordinates, say (x,y,z). A 4D point has four coordinates, say (w,x,y,z). The 3D cube has eight vertices at:

(+/-1, +/-1, +/-1)

Therefore a very natural extension into four dimensions would be:

(+/-1, +/-1, +/-1, +/-1)

For a total of sixteen vertices. To look at it another way:

(1, +/-1, +/-1, +/-1)
(-1,+/-1, +/-1, +/-1)

That is, at w=1 we get a cube, and at w=-1 we get another cube. In fact, if we take a "slice" of our hypercube, we get a 3D cube. Compare to taking a slice of a 3D cube, where you get a square (a 2D cube, if you will).

This is demonstrated when the code first starts up -- the program "grows" a cube from 0D -> 1D -> 2D -> 3D -> 4D. At the 4D stage there is a smaller cube inside of a larger cube, with cubes in-between the two. (If you are curious as to how I did the "growing", see the code description below for a few details).

Next, as the cube begins to rotate, it "folds in" on itself (or, if you like, it unfolds!). Rotations are no different than they have always been. To do a 3D rotation, recall that the object is rotated in the x-y plane, the y-z plane, and the x-z plane. To rotate in the x-y plane by an angle phi:

```
xnew = x*cos(phi) - y*sin(phi)
ynew = x*sin(phi) + y*cos(phi)
```

Well, any two coordinates form a plane, so in four dimensions there are just twice as many planes to rotate in. In particular, the program does rotations in the usual planes (x-y, y-z, x-z) and also does a single rotation in the w-x plane, that is,

```
wnew = w*cos(phi) - x*sin(phi)
xnew = w*sin(phi) + x*cos(phi)
```

I didn't feel any great need to rotate through extra planes involving the w-coordinate (the w-y and w-z planes). When phi=90 degrees, or 180 degrees, notice that the coordinates trade places, then go to their negatives. This means that as phi is increased, in essence the inner and outer cubes are going to change positions, and this then explains the unfolding that is seen on the screen.

The R/S key goes into 3D mode by zeroing out the angle increment for the w-x plane. In effect, the 4D rotation is frozen. The F4 key zeros out the x-y, y-z, and x-z angle increments, leaving only the w-x rotation. F4 followed by R/S will therefore freeze the image completely -- use D or 4 to get it going again.

There is still the issue of visualizing a 4D object. This should not be surprising -- after all, we have all seen 3D objects drawn on a 2D computer screen (or a 2D piece of paper). If we can get from 3D to 2D then we ought to be able to get from 4D to 3D (and from there into 2D). Recall that a 3D projection draws a light ray from the object, through a little pinhole located at the origin, and finds the intersection with a piece of film located at z=d, a constant:

```
L = t * (x1,y1,z1) is my light ray, so t=d/z1 gives the
                   intersection with the film of a ray from
                   the point (x1,y1,z1) passing through the
                   origin.
```

So this is very easy to extend into 4D -- simply project from 4D into 3D through the origin:

```
L = t * (w1,x1,y1,z1)  let t=d/w1

-> L3 = (d, d/w1 * x1, d/w1 * y1, d/w1 * z1)
```

The x,y,z coordinates are then projected from 3D into 2D, again through the origin. This gives a "perspective" view of the 4D object.

Now, what is the 4D analog of a tetrahedron, or an octahedron? I reasoned them out by trying to think of what 3D objects I could derive starting from a cube. That is, taking a cube, and cutting away pieces of it. For instance, to do the 14-sided guy, simply take the midpoint of each line segment on the cube -- this has the effect of cutting off the corners of the cube. By defining things in this way, it is fairly

straightforward to extend the objects into four dimensions. (I was happiest to realize how to do a tetrahedron). See the file objects.s for more details on the individual objects. Naturally each has some similarity to the cube: there is an inner object(e.g. a tetrahedron) and an outer. The two are connected, and each set of connections forms another object, so that, for instance, there are tetrahedrons in-between the inner and outer tetrahedrons.

Finally, to help in visualizing the objects, I stuck a dotted line capability in. The dotted lines in general connect the "inner" and "outer" 3D objects -- turning them off lets you then see the two objects interact. (The third object was mighty impressive-looking before I added these guys! :)

@(A): The Code

Now, it is my considered opinion that the code is awfully well documented, so there isn't too much to say, but a few general things are worth mentioning.

"Growing" the points is really easy -- simply start each coordinate at zero, and gradually increase it out to its final value. By doing this first with the x-coordinates, then the y-coords, then z, then w, the cube grows a dimension at each step. I don't do anything fancy with the other objects -- all coordinates are grown equally, so the objects grow outwards from the origin (as opposed to some sort of zoom effect).

Each 4D character is a 12x12 character grid, which gives a 96x96 pixel drawing area, and takes up the first 144 characters. Each 3D character uses a 5x5 character grid, giving 40x40, and taking up the next 4*25=100 characters, for a total of 244 so far. In eight of the remaining 12 characters are four patterns and their EOR #$FF complements, which are used in the background tilings and are used indirectly in the pattern fills.

Since the final x-y coordinates can range from -48..48, this places a restriction on the initial values for the coordinates. For purposes of accuracy and such coordinates must of course be scaled, so that while a coordinate like (1,1,1,1) is convenient for thinking, a coordinate like (16,16,16,16) is much better suited to the implementation -- that is, the original coordinate scaled by a factor of sixteen or so. The table range restricts this scaling factor: the 4D coordinate with largest length that I use is (1,1,1,1), which has length 2. Thus, after rotation, it is possible that it will lie on an axis with coordinate, say (2,0,0,0). Since coordinates must not exceed 48 in the implementation, this suggests a scaling factor of 24.

As a practical point, the points never really hit this maximum, so in principle a larger scaling factor could be used. Alternatively the projection routine can pick up the slack, which is what dim4 uses.

The first smart thing I did was to ditch the old method of computing rotations. Instead of calculating a big rotation matrix, I calculate some big tables of f_x (s) = x*sin(s), and let the angle s range from 0..127. To get a table of cos(s) I simply periodically extend the sine table by copying the first 32 bytes of the table into the 128-159 positions -- cos(s) is thus sin(s+32). (I take advantage of the fact that sin(s) and cos(s) are related by a factor of pi/2. Were I smart I would have taken advantage of the reflection symmetry of sin/cos, and saved another 64 bytes. Oh well.)

This then leaves 96 bytes for a projection table, which is just what I need for the 4D object. Thus I can mash tables of x*sin(s), x*cos(s), and my projection table of f_x(z)=d*(z-z0) * x into a single page. This page is then extended from $6000 to $C000, i.e. giving 96 tables, for a total of 24k. Accessing the tables is now trivial: store x+$60 in the high byte of a zero page pointer, the low byte contains the offset into the table (0 for the sine table, 32 for the cosine table, and 160 for the projection table), and do an LDA (ZP),Y to get the right value.

Thus rotations and projections are now very fast and very compact. Note that it isn't really necessary to generate a complete table of sines and cosines. For instance, 12k of tables (or 6k or whatever) could be used, and the final result simply multiplied by two, or four. Even though the final coordinates might range from -48..48, calculations don't need to be done using the full range.

The line routine is the good 'ol chunky line routine from the last cube3d program. It of course had to be modified to work with the two buffers and such. I removed a bunch of really redundant code that was in there (REALLY redundant), especially in the actual drawing part (macros

XSTEP and YSTEP -- lines are commented out with a '*'). I also added a dotted-line capability (it only takes a few extra instructions), to make things easier to see.

Only a single 3D object is actually drawn -- the others are generated via symmetry (reflections through x=0 and y=0). Since the 3D objects are drawn on a much smaller grid, they need to be scaled down a bit. Instead of writing separate routines to deal with the 3D and 4D objects, I simply set the 4D coordinate of each point in the 3D object to some appropriate number. Recall that in a 3D projection, the farther away from you the object is, the smaller it gets. This is the same idea -- the object is pushed down the 4D axis, and this has the effect of shrinking the object upon projection.

You may have noticed that the 3D objects tend to avoid the center of the screen -- this is a consequence of the random number generator I coded up (and did not test for spectral properties or anything like that :). Originally I was going to place things in a random row and column, but then things just clumped along a diagonal line :). I will also say that the SPLAT routine caused me many days of headaches -- whose idea was it to put color memory so close to a CIA? :)

One thing I had to prune out was a routine which draws circles as the sine/cosine tables are being set up. It is kind of neat and gave me something to watch while the code was setting up, and also was a check that the trig tables were being set up correctly. Anyway, all it does is to draw concentric circles of progressively larger radii, for a sort of tunnelish-looking thing I suppose.

There is a little "failsafe" in the projection routine. If coordinates are out of range (greater than 96 or 40) after projection, they are set to the origin. At least one of the objects screws up from time to time (the octahedron is the main culprit I think), and I think what happens is that the line routine thinks it needs to draw a lot more points than it really needs to. So it happily moves along sticking bytes into the trig/projection tables, and even makes its way up to VIC, SID and the CIAs! Once, it actually started pegging the SID volume register or something, because there would be a periodic loud ticking from the speaker. Eventually the code just grinds to a halt or else completely hoses the system -- hence, the failsafe :).

Finally, the very first lines of the code redirect the BASIC vector at $0302/$0303 and JMPs to the NMI RS/RESTORE routine (although a BRK would probably have sufficed). This is the only way I could get the code to work with the cruncher -- without it, the program goes into "IRQ lock". Crossbow of Crest suggested that ABCruncher does not put a CLI at the end of its crunching routine, and that this can cause problems, most notably with the CIAs.

It took 10-15 hours to get things to crunch and work correctly. In hindsight, I can think of a bunch of things that could have been easily done to make it work, but at the time I was sure relieved when it finally got down to 4095 bytes. Moral: A little thinking early on saves massive time and effort down the road.

@(A): The Music

Finally, a word about the music. Originally I was going to construct a series of chords which I could modulate between in a fairly flexible way. I was then going to break up the chords in a nice way and move between them randomly. But then it occurred to me that I already knew a piece of music which was a series of broken chords and sounded infinitely more cool than anything I was going to accidentally write, so I used it instead. Even better, they are four-note "chords", broken into four groups of four notes each -- too good to pass up. Notes are looked up in a frequency table, thus on my PAL 64 the music gets transposed to a different key (in addition to playing at the wrong speed :).

I do not necessarily recommend using the routine as a model for doing IRQ interrupts -- I had many problems with "IRQ lock", where an IRQ is continuously latched, and consequently is constantly running the routine. I still do not understand what is happening, nor do I have a solution.

@(A): Memory Map

$0F00-$0FFF	Starting sine+projection table
$1000-$2257	Code
$3000-$4000	Character set
$6000-$C0FF	Sine, cosine, and projection tables

$C100-$CFFF   Misc. variables and tables

@(A): Contents of dim4.lnx

Note: the code is available in this issue of Commodore Hacking (Reference: code, SubRef: democode), on the Commodore Hacking MAILSERV server (Reference: code), and at http://www.msen.com/~brain/pub/dim4.lnx

dim4	Submitted entry for 4k demo contest
dim4.text	This file, in PETSCII format
dim4readme-runme	Obvious
dim4.names	Linker name file to use with Merlin 128
main4.s	Main code for dim4
objects.s	Code to define/set up objects
graphics.s	Various graphics routines (lines, fills, etc.)
music.s	Init and main IRQ music routine

==========

@(#)cpu: Exploiting the 65C816S CPU
by Jim Brain (j.brain@ieee.org)

@(A): Introduction

For a CPU architecture that can trace its roots to the mid 1970's, the 65XX line has proved very successful. Finding its way into flagship systems such as Commodore, Apple, Atari, and other lesser known units, the CPU has toiled away for years in the single digit megahertz speeds. Programmers across the world have analyzed the CPU to death and documented every last one of its "undocumented" opcodes. Ask a "coder", and he or she will rattle off the cycles it takes to do an immediate load or an absolute store. In short, the CPU is road tested and well known.

However, how much do you know about its "children"? Yes, in the 1980's, while Commodore was busy tinkering with the NMOS version of the CPU designed by Chuck Peddle, Bill Mensch, and the ex-Motorola 6800 design crew, Bill Mensch started a new company, Western Design Center, and redesigned the 6502 to use the newer and faster CMOS fabrication process. In addition to the new 65C02, Mensch designed an upwardly compatible 16 bit brother, the 65C816. Although both were offered to Commodore, only the 65C02 was used and only in the never produced CBM Laptop computer. Apple, however, used the 'C02 in later models of the Apple II line and placed the 65C816 at the heart of the Apple IIGS system.

Although Commodore never took advantage of the WDC CPUs, third party products have offered their speeds to the Commodore community. Early models like the TurboMaster and TurboProcess offered 4 MHz speeds to the Commodore 64 owner, while newer products like the FLASH8 offered 8 MHz speeds. The fastest offering thus far is the CMD SuperCPU, offering speeds of 20 MHz to the Commodore owner. Of these, the TurboProcess, the FLASH8, and the CMD SuperCPU all use the 16 bit CPU, the 'C816.

Since the 'C816 is available now to the Commodore user, and with the SuperCPU poised to provide software compatibility never before achieved, it is likely that more and more Commodore applications will run on 'C816 equipped machines. So, why should the Commodore software developer care? Sure, the 65C816 will run 6502 based applications in 6502 emulation mode at substantial speed increases, so developers can opt to continue writing 6502 based applications. While I encourage developers to always provide 6502 based versions of applications when possible, there are useful features available only in the Native mode of the 65C816. This article describes some of these features and how to utilize them.

@(A): Disclaimer

The following information is based on following resources:

- Data Sheets on the 65C816S, Western Design Center
- _Programming the 65816_, by David Eyes and Ron Lichty, 1985, Western Design Center.
- A beta version of the SuperCPU 20 MHz accelerator from CMD.
- A beta version of the Super Assembler (SAS) 65C816 Assembler, by Jim Brain, Distributed by CMD.

Most of the following information is system independent, but any information specific to the CMD SuperCPU is preliminary and subject to change.

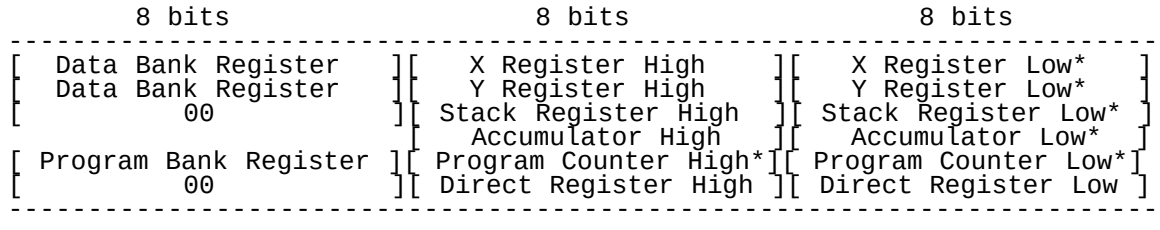
It is not the intention of this article to detail all the possible 65C816S opcodes nor their addressing modes. It is also not the intention of the article to describe the operation of the SAS assembler. For more information on both of these products, please consult the manuals listed above.

@(A): Diving Right In

As this article is geared toward the programmer, we're going to dive right into the new features. Commodore World issue #12 has an overview of the CPU for those just arriving on the scene. For those who know an index register from an accumulator, read on

@(A): Overview of Registers

One of the features of operating in Native mode of the CPU is the enhanced set of registers available to the programmer. They are also key to explaining the other features of the CPU. So, let us go over the new register set:



* Original NMOS 65XX register set

These registers are referred to in the remainder of the article by their acronyms, as follows:

Data Bank Register	(DBR)
Program Bank Register	(PBR)
X Register High	(XH)
X Register Low	(XL)
Stack Register High	(SH)
Stack Register Low	(SL)
Y Register High	(YH)
Y Register Low	(YL)
Accumulator High	(B)
Accumulator Low	(A)
Program Counter High	(PCH)
Program Counter Low	(PCL)
Direct Register High	(DH)
Direct Register Low	(DL)

In addition, the 16 bit combination of B:A is called C, the 16 bit X and Y registers are called simply X and Y, the 16 bit Direct Register is called simply D, and the 16 bit Stack Register is called S.

One more register requires discussion before we can delve into programming the '816: the Status Register (P)

Bit:	Description
7	N flag
6	V flag
5	1 in Emulation mode
	M flag in Native mode (memory select bit)
	0 = 16 bit accumulator
	1 = 8 bit accumulator
4	B flag in Emulation mode
	X flag in Native mode (Index Register Select)
	0 = 16 bit X and Y registers
	1 = 8 bit X and Y registers
3	D flag
2	I Flag
1	Z flag
0	C flag
	E flag (Emulation flag) (Can not be accessed directly)
	0 = Native mode
	1 = Emulation mode

It is important to note that there are 3 more flags available in the Native mode version of the status register. Since there were 7 flags used before, how did WDC squeeze in the extra flags? Well, the E flag cannot be accessed or seen in the status register. The only way to

change it is to set up the C flag to the intended stats of the E flag and issue the eXchange Carry and Emulation flags (XCE) opcode. Another flag, M, takes the place of the static 1 state in the old status register. M controls the length of the accumulator. The last flag, the X flag, controls the length of both index registers. Note that this flag takes the place of the B flag. Thus, the B flag is unavailable in Native mode. Since the B flag is used to determine whether a hardware IRQ or a software BRK opcode caused an IRQ interrupt, the Native mode provides separate interrupt vectors for BRK and hardware IRQs.

The X and M flags are especially important in Native mode, so much so that each programmer will become intimately familiar with these flags. When a register is selected to be 8 bits wide, it emulates the operation of the register in Emulation mode. However, when the register is flipped into 16 bit operation, its length doubles everywhere. For instance, a push of the accumulator with the M flag reset causes 2 bytes to appear on the stack. Likewise, an immediate load of the accumulator will require a 3 byte instruction: one for the opcode, and a 2 byte operand. This opens up one of the nastiest gothcas on the chip, but we'll detail this later in the article.

@(A): More Memory

As you may be aware, the Native mode of the '816 allows the programmer contiguous access to up to 16 megabytes of RAM. This access doesn't involve tricks such as DMA, page flipping, or RAM "windows". At any given point in time, an application can access a memory location and request a memory location more than 64 kB higher in the next instruction. In order to access the new memory locations using standard 6502 addressing modes, the new DBR and PBR registers have been added. The PBR serves as the 3rd byte of the PC, allowing code to run at any location in memory. The DBR register functions as the 3rd byte for memory accesses in addressing modes like absolute mode. Of course, there are restrictions, like the inability to execute code that crosses a 64kB boundary, but these restrictions can be overcome, as you'll see below.

For clarity, we will refer to the 3rd bytes of an address as the "bank", and refer to the 2 lowest bytes of an address as the offset. Alternate names include "segment" and offset, but that naming scheme was previously used with the Intel 80X86 CPU line and carries with it many bad connotations.

Since memory addresses can now be 3 bytes wide and contain 6 hexadecimal digits, an obvious representation would be $xxxxxx. However, many '816 references write the address as a two part quantity, with the bank register and the 16 bit offset separated by a colon, ":". Therefore, $xxyyyy and $xx:yyyy are equivalent. In this article, the former notation is used for emphasis and because ":" notation also brings up bad connotations from Intel 80X86 CPU line.

@(A): Increased Stack

As the S register is now 16 bits wide, the stack can now reside in all of bank 0, giving the programmer 64 kB of stack area. As well, the S register can be set to any location in bank 0. This allows one to start stack from any non-aligned page in bank 0.

@(A): Enhancements to Old Addressing Modes

Even though the '816 supports the traditional 14 addressing modes of the 6502, it extends some of them to handle the extra features in the '816. Note that the opcodes and parameters have not changed for these addressing modes; rather the way the CPU treats them differs slightly. Of special note is the term "zero-page", which has been expanded into "Direct Mode". Let's take a look at what changes you can expect.

@(A): Absolute Modes

In the 65XX CPU, modes such as absolute and its indexed siblings each could access a memory location in the 64 kB memory map. In the '816, these modes are now capable of accessing memory above and beyond 64 kB. When accessing memory, the DBR register is prepended to the address being accessed, thus forming a 24 bit effective address. When transferring control, the PBR register is prepended. Thus, if the DBR contains a $05, the following:

```
   af ff ff     lda $ffff
```

would load a value into the .A register from $05ffff. If the M flag is set to 16 bit mode, the 16 bit value in $05ffff and $060000 will be

loaded. If the M flag is set to 8 bit, only $05ffff will be loaded.
Notice that this example also shows "temporary-bank-incrementing".
While loading a 16 bit value with the instruction above, the DBR is
"temporarily" incremented to allow access of data from bank $06. The
actual DBR is left unchanged, so the next instruction will find the DBR
back at $05.

You'll rarely see such bank changes when accessing data as above, but it
is common when using indexing modes. With the DBR at $05, executing:

```
   a2 ff ff     ldx #$ffff
   bd ff ff     lda $ffff,x
```

will load values from $06fffe and possibly $06ffff, depending on the
size of the accumulator.

When using absolute mode on opcodes like JMP and JSR, the PBR register
is used to form the 24 bit effective address. Unlike the DBR, the PBR
does not exhibit "temporary-bank-incrementing". It simply rolls over
within the same bank. Keep that in mind.

@(A): Direct Modes

To enhance the capabilities of the '816, the CPU offers Direct Mode,
which is a superset of "zero-page mode". Basically, all z-page opcode
operands are added to the D register to form a 16 bit effective address.
This allows using the entire bank 0 as effective z-page memory. With
the D register set to $0200, executing:

```
   a5 10        lda $10
```

would load the accumulator from $000210 (and possibly $0211). Direct
mode is not allowed to increment into bank 1. If the above instruction
is executed while D = $ffff, the accumulator would start accessing data
from $000009. This highlights an important yet subtle change. No
longer is lda $10 guaranteed to access data from $000010. It will
access data from D + $10.

Indexing changes little with respect to Direct mode. After the D
register is added to the 8-bit offset, the appropriate register is
added, and the effective address is normalized to fall within bank 0.
There is no way to reference outside bank 0 in Direct mode. Even if
index registers are set to 16 bit mode and hold $ffff, the instruction
will access bank 0.

@(A): Direct Indexed Indirect Mode

Most programmers forget, but this mode executes in two parts. Now, it
becomes important. In the first part, the 8-bit offset is added to the
D register and then the X register. The result is normalized to 16
bits, and two values are accessed from bank 0. The second part takes
those two bytes as the effective address, and PREPENDS the DBR register
to form a final address. In this way, you can access memory outside
bank 0 with this mode, but you must store the address to access in bank
0. Read that sentence again.

@(A): Direct Indirect Indexed Mode

Like its relation above, this mode work in two parts. In part 1, the 8
bit offset is added to the D register and normalized to 16 bits. Two
bytes are accessed from bank 0, and then the 16 bit value returned is
appended to the DBR to form a 24 bit effective address. In part 2, the
Y register is added to this effective address to form the final address
for access. As above, part 1 cannot access outside bank 0, but part 2
can.

@(A): Stack Mode (Implied)

Usually lumped in with the Implied addressing mode by most 6502
developers, stack mode has changed to accommodate the new widths of the
registers. Depending on the width of the register, stack operations
will push and pull either 1 or 2 bytes. This can cause problems if you
push a 16 bit register and try to pull it off as an 8-bit register.
Caveat Emptor.

@(A): Immediate Mode

In Emulation mode, immediate mode was simple. You specified an 8 bit
immediate value to be loaded into a register. In Native mode, however,
registers can be 16 bits. Everyone knows the opcode can do an immediate
8-bit load, but what opcode performs a 16-bit immediate load? Answer:

the same opcode! If the register is set to 8 bits via the X or M flags, the immediate load on that register will pull in 8-bits. If the register is set to 16 bits, the instruction will load a 16 bit value. The effects of this change are monumental. An 8-bit immediate load requires 2 bytes, while a 16 bit load requires 3.

This presents some problems. Since neither the opcode nor the mnemonic differs between the two forms, the assembler cannot tell which form is required from context. The develop must tell the assembler which form to use by use of assembler directives. However, this doesn't guarantee success. The developer must ensure that the flags are set correctly before executing an immediate load of any register. Improper settings will either cause the instruction to pull the next opcode into the high byte of the register or treat the high byte of the intended register value to be executed as an opcode. In my biased opinion, this is severely shortsighted. I would rank this as the number one bug that '816 developers will face. However, simple macros employed in your assembler can help minimize this problem.

@(A): New Addressing Modes

The 816 can utilize all 14 original addressing modes of the 65XX line, and adds 10 more for a total of 24. The new addressing modes are as follows:

```
*  Absolute Long                      al
*  Absolute Long Indexed              al,x
*  Absolute Indirect                  (a)
*  Absolute Indexed Indirect          (a,x)
*  Direct Indirect Long               [d]
*  Direct Indirect Long Indexed       [d],y
*  Stack Relative                     d,s
*  Stack Relative Indirect Indexed    (d,x),y
*  Relative Long                      rl
*  Block Move                         xyc
```

Let's take each under consideration:

@(A): Absolute Long Mode, Absolute Long Indexed Mode

These modes allow a programmer to access a fully qualified memory location without using the DBR. The benefits include pulling data from one bank to store in another without constantly changing the DBR. The disadvantages include the extra size of the instruction. The long forms of these two absolute modes takes an extra byte in memory and an extra cycle to load into the CPU. Note that only the X register is supported as an index register for absolute long indexed mode.

@(A): Absolute Indirect Mode

This mode, denoted as (a), functions similar to (d), but does not require the D register. The locations in bank 0 specified as the operand in this mode are accessed, and the results form the lower 16 bits of the effective address. The PBR is prepended to this address to form a final 24 bit address. The PBR is used since only the JMP opcode uses this mode.

@(A): Absolute Indexed Indirect Mode

This mode, denoted as (a,x), functions similar to (d,x), but the D register is not involved. Thus, as with the (a) mode above, these modes are to direct modes as absolute long modes are to absolute modes.

@(A): Direct Indirect Long Mode

This mode functions similar to Direct Indirect Mode, except that in part 2, all three address bytes are pulled from memory. The DBR and PBR are not involved, but the D register is used. If locations $10 - $12 contained $10, $11, $12, and the D register contained #08, then:

```
    a7 08       lda [$08]
```

would load the accumulator with data starting at $121110. Do you see how that works? The D register is added to $08, and the result ($10) is accessed to fetch the 24 bit memory address, in low byte order.

@(A): Direct Indirect Long Indexed Mode

As Direct Indirect Indexed Mode extends Direct Indirect Mode, this mode extends the above mode by adding the Y register to the effective address

pulled from bank 0. Note that even though this mode uses a fully qualified 24 bit address (no DBR or PBR involved), it can still increment into the next bank to access memory. Thus if we use the above example, and Y = $ffff, executing:

```
    b7 08       lda [$08],y
```

will fetch the accumulator starting at ($121110 + $ffff, or $131109).

@(A): Stack Relative Mode

Denoted as "d,s", this mode is completely new to 6502 programmers. It starts off the set of modes that work with the S register (Stack Pointer). As the Stack Pointer is now 16 bits in width, the stack can fill all of bank 0. Although 65XX programmers have traditionally used stack locations only for saving return addresses from JSR and interrupt sources, the '816 allows one to store data on the stack. In this mode, the 8 bit operand is added to the S register and normalized to 16 bits. Memory in bank zero is accessed starting at this effective address. Since the S register points to the next location to hold data, executing:

```
   a3 00        lda $00,s
```

would prove meaningless, unless you wanted to get the last byte pulled off the stack. This mode allows one to access the last 255 bytes off the stack in the order they would be pulled off.

@(A): Stack Relative Indirect Indexed Mode

By far the most complex Addressing Mode to understand in the '816, this mode can be used to access data referenced by pointer values on the stack. Denoted as "(d,s),y", the effective address formed by the 8 bit operand and the S register is normalized to 16 bits and 2 bytes are accessed. The resulting 16 bits are appended to the DBR register and the 24 bit effective address is added to the Y register to form a final 24 bit memory address. This can be used to access data passed by "reference" (not the value, but the pointer to the value, is stored in the stack).

@(A): Relative Long

Only one opcode uses this addressing mode, Branch Long (BRL), and it fulfills the desire of every 65XX programmer to have a relocatable jump instruction. Unlike normal branches, BRL can cross page boundaries. However, BRL is constrained to the current bank. It cannot cross banks. Although viewed as a disadvantage, this presents a few possibilities. If a programmer was at $xxff00 and wanted to jump to $xx0000, he or she can use BRL, even though the offset appears wider than $32767, the maximum offset for BRL. In actuality, the assembler computes a branch to the next bank, which is only 256 bytes away. The CPU negates the bank increment, thus forcing execution to begin at the current bank.

@(A): Block Move

Along with stack relative indirect indexed mode, this mode is complex. However, unlike its stack counterpart, this complex mode is easier to understand. Denoted as xyc, this mode allows the programmer to quickly move areas of memory from one bank to another. An example will prove helpful:

```
   a2 00 20     ldx #2000
   a0 00 30     ldy $3000
   a9 ff 0f     lda $0fff
   44 02 01     mvn $01,$02
```

Basically, we are moving $1000 bytes from $012000 to $123000. The X register holds the offset into the source bank; the Y register holds the offset into the destination bank. The accumulator holds the number of bytes to move MINUS 1. Remember that. The opcode Move Negative (MVN) takes the source bank and the destination bank as operands. The only opcodes that utilize this mode are MVN and Move Positive (MVP). MVP assumes the X and Y registers hold the top of the data areas to move, while MVN assumes the opposite.

@(A): Hints And Tips That Will Decrease Your Stress

Writing Machine language applications on any platform is bound to create stress in your life, but this section is presented to make the programmer aware of some "gotchas" in the '816. Here goes.

@(A): Initialization

To switch the processor from emulation mode into native mode, perform the XCE (eXchange Carry with Emulation) mnemonic with the carry bit reset:

```
   18           clc
   fb           xce
```

The next thing to do is determine the initial size of your registers. The 816 can use any register as 8 bits or 16 bits. By default, the registers are 8 bit, but just to make sure:

```
   c2 30        rep #%00110000  ; set index and acc to 8 bit
```

By stuffing $30 into the processor status, we are setting both the X and M flags to 8 bit.

At this point, it should become obvious that if the programmer wishes to flip between 8 and 16 bit modes many times, macros need be employed to do this quickly and painlessly.

Now, the development can begin.

@(A): Register Usage

Never underestimate the power of immediate mode to mess your program up. If at all possible, switch to one size of registers and stay that way. If that can't be accomplished, thoroughly document where you changed the size of either set of registers. Remember, the assembler cannot trace program execution, so don't assume the assembler will fix everything up for you. On the preliminary version of the SAS assembler, the opcodes to instruct the assembler to alter immediate mode behavior are:

```
   .inl   ; INdex registers Long
   .ins   ; INdex registers Short
   .acl   ; ACcumulator Long
   .ins   ; ACcumulator Short
```

What happens when you change a register's size? Well, let's treat the index registers and the accumulator separately. When changing from 8 to 16 bits, the index registers are simply extended by padding with zero. When the index registers are changed from 16 bit to 8 bit, the high byte of each index register is lost and forced to zero. On the other hand, the accumulator is actually made up of two 8 bit registers. When changing from 8 to 16 bit, the accumulator's high byte becomes the value in the hidden B register. When moving from 16 to 8 bit, the high byte of the accumulator is forced to zero, but the B register is left intact. Thus, changing from 8 to 16 bit and back to 8 bit won't affect the accumulator, but it will force the high bytes of the index registers to zero.

When using the MVN and MVP opcodes, the size of the index registers make a difference. If set to 8 bits, one can only transfer memory from page 0 of any bank. However, unlike the index registers, MVN and MVP treats the Accumulator as 16 bits wide, regardless of the state of the M flag.

Also, what length the registers are set to determines how many bytes are pulled or pushed during register to stack operations. Remember that when pushing from one location and pulling in another.

To execute emulation mode code in Native mode, simply set all registers to 8 bit widths, load the D register with 0, and load the stack with $01ff. By manipulating the DBR and PBR, you can execute up to 256 emulation mode programs, while at the same time using this method.

In the '816, there is no need to use a register to zero out memory. The "stz" opcode can be used in the same manner as "sta" to zero out memory. Note that, like "sta", stz will store a single or double byte 0 depending on the state of the M flag.

Remember that, when the accumulator is set to 16 bits, the BIT instruction no longer copies bits 6 and 7 to flags in the P register, but bits 15 and 14.

@(A): Timing

Beware of absolute long addressing. It takes 1 more byte and 1 more cycle to utilize. Use it sparingly.

In Native mode, there is no penalty for crossing a page in memory. This

should allow some programs to actually run faster in Native mode.

BRL can be used at any location a JMP would be used. The advantages include self-relocatable code, but the disadvantages include an extra cycle for execution.

By now, you have noticed that MVN and MVP provide a fast way of moving data areas. However, they can also be used as a fast fill. Simply store the fill pattern into the first address of the memory area, load the X register with the start of the fill area. Load Y with the start plus the length of the fill pattern. Load A with the size of the fill area minus the fill pattern size minus 1. Then, do a mvn h,h, where h is the bank you want to fill. Any size pattern can be used.

@(A): Stack Instructions

Many of the added instructions in the 65816 deal with enhanced stack operations. In addition to the S register and accumulator pushes and pulls, you can now programmatically push all the registers except PC onto the stack, and pull all but the PBR and PC register off the stack. Note that some registers have variable sizes, while others are fixed in width. The breakdown is as follows:

```
Fixed at 8 bits:   DBR, PBR, P
Fixed at 16 bits:  D, S, PC
Variable:          X, Y, A
```

In addition to using the stack to save and restore data and registers, addresses can now be programmatically pushed to and pulled from the stack. The following opcodes are available:

```
   f4 34 21     pea $1234   ; Push Effective Address 1234 on stack
   d4 21        pei ($21)   ; Push Effective Indirect Address at D + $21
                            ; on stack
   62 e1 7f     per DATA    ; Push Effective Relative Address on stack
                            ; when executed, the address PC + $7fe1 will
                            ; be pushed on the stack.  Useful for
                            ; determining data area locations in
                            ; relocatable code
```

@(A): Transfer and Exchange Operations

When using the accumulator as an 8 bit register, the special hidden B register can be used as a "hidden" register. Move it into focus with the XBA (Exchange B with A) opcode. Note that this is a swap, not a transfer.

Transferring between same size registers in unambiguous. Transfers between different size registers is tricky. If the accumulator is set to 8 bits, only that much can be transferred in, but all 16 bits will be transferred out to any 16 bit register, regardless of the state of the M flag. If an index register is set to 8 bits, only that much will be transferred in or out.

@(A): Addressing Modes

Beware of Direct Mode. Any address that can be represented by a single byte will be assembled into Direct Mode. Sometimes, absolute zero page addresses are desired. Use the "!" directive to force absolute addressing.

Beware of Direct Mode II. Remember that zero-page is no more. If you intend to use z-page as before, remember to set D to $0000.

@(A): Miscellaneous

This article is presented to new CMD SuperCPU programmers. Whether you write SuperCPU applications in Emulation or Native mode, however, you will find the following information helpful:

The SuperCPU contains a set of registers to control operation of the unit programmatically. These new registers are located in "mirror" locations of the VIC-II (6567/6569) IC. On a stock system, these locations return $ff when read, and writing these locations does not affect RAM under the "mirror" locations while I/O is switched in. These locations are considered relatively "safe" and have been chosen to contain these important CMD SuperCPU registers:

Location	Purpose
$D074 (1)	GEOS Optimization (mirror VIC bank 2, $8000-$BFFF)

$D075 (1) VIC bank 1 Optimization (mirror $4000-$7FFF)
$D076 (1) BASIC Optimization (mirror $0400-$07FF)
$D077 (1) No Optimization (Default; mirror all memory)

$D07A (2) Software Speed Select - Turbo Off (1 MHz)
$D07B (2) Software Speed Select - Turbo On (20 MHz)

$D07E (3) Hardware Register Enable
$D07F (3) Hardware Register Disable

Notes:
(1) Write only, hardware registers must be enabled to access location.
(2) Write only, may be accessed with hardware registers enabled or
    disabled, but does not over-ride hardware Speed switch.
(3) Write only.

The first 4 locations specify how much and what areas of RAM will be synchronized between the SuperCPU and on-board RAM images.

These registers have been created using a "sandwich" method that minimizes irregular operation due to memory fills. As such, each register has a "shadow" that falls two bytes away from the register itself. During a memory fill, a fill might turn off fast mode by writing to $d07a, but any access to $d079 or $d07b will turn fast mode back on. This would cause the machine to operate in the wrong state for at most one instruction period. Only one address of each register is documented, as the shadows of each register should not be used for program development.

To utilize the above registers, the programmer need simply to write a value into the appropriate location. In the tradition of CMD, it is not relevant what value is stored at a location. Rather, that a memory write occurred at that location suffices.

In addition to these outlined registers, there are additional "bit-mapped" registers in the VIC-II register map that signal the state of the SuperCPU hardware and software. These flags are read only when hardware registers are disabled, and read write when the hardware registers are enabled. More information about these flags and their locations will be included in the SuperCPU Developer's Guide. Programmers should use and modify these flags with extreme caution.

In addition to the above registers, there are two pages of RAM present at $d200 and $d300 on the SuperCPU. Although this memory is present, it is dedicated for SuperCPU use and should not be otherwise utilized.

@(A): Conclusion

Well, there you have it. I am learning something new about this CPU every day, and some of these modes still baffle me. However, I hope that each of you takes an interest in developing '816 applications, as the possibilities are endless. Just when you thought you had the 65XX line all figured out...

==========

@(#)html: Using HTML on the Commodore, Part 1
          by Jim Brain (brain@mail.msen.com)

Note: Due to the recent relocation of myself and my family, I am behind on the development of the HTML viewer for the Commodore system. Therefore, this article will not focus on the actual viewer. With the development below 50% complete, the modules are subject to change. Describing them now would only confuse issues.

@(A): Introduction

HTML. This simplistic acronym, unknown to most people before 1993, now forms the heart of discussions. Its status is secured, as employers ask how much "experience" one has with it, and resumes commonly include it. A quick tally in any technical magazine reveals hundreds of references to it, and trips to the bookstore yield mountains of titles referring to it.

Most Commodore owners have a few questions about this acronym. First, what is it? Second, why should I care about it? In this series of articles, I will try to answer both questions to your satisfaction.

To answer the first question, let's step back to explain the World Wide Web (WWW). This explanation is not designed to replace more thorough treatments of the subject. In 1991, while working as a researcher at

the CERN laboratory in Switzerland, Tim Berners-Lee developed a hypertext information retrieval system that allowed researchers at the lab to design informative online "presentations" of their work. In each presentation, a researcher could reference a document or presentation located elsewhere on the lab-wide network of computers. This reference was "live", meaning that a person could select it from the document and immediately view the referenced document. Thus, a matrix of related documents were created to interconnect the researchers' work.

In an effort to offer the researchers great latitude in presenting their works while retaining some standard in layout, Berners-Lee found simple ASCII text an inadequate presentation method. Clearly, a document formatting procedure, or "markup language" was needed. However, Berners-Lee found that popular document markup languages did not support the concept of referencing, or "linking" between documents in a standard and non-proprietary way. After looking past popular approaches like Windows help files, troff, TeX, and Rich Text Format, Berners-Lee found a standardized markup language that would support links and provide flexibility in creating documents, yet retain some semblance of commonality. The language was the Standard Generalized Markup Language (SGML).

SGML in itself was derived from an IBM specific markup language called Generalized Markup Language (GML). After some minor changes, the IBM GML specification became standardized. SGML, though, represents more than a simple formatting schema. SGML allows one to create multiple derived markup languages off the SGML base, and a suitable program can interpret each derived language independently. Thus, HTML functions as a derivation of SGML.

Berners-Lee created the original specification for HTML while working on the WWW framework. Since mid 1993, when the first graphical HTML viewer arrived from the University of Illinois, the HTML specification has been revised and updated at least 4 times, but remains an SGML derived language.

@(A): The Basics of HTML

HTML, like most formatting or document markup languages, allows the document creator to insert special labels, or "tags" into the document, which the language processor can parse. The language processor then converts these tags into the special formatting options they represent. In a simplistic markup language, one might place an asterisk "*" next to any word to be highlighted. As the "marked up" document is read and parse by the language processor, the resulting output would highlight each word preceded by an asterisk. The asterisk itself would be stripped from the resulting display, as it does not form part of the document itself. In much the same way, HTML allows creators to insert HTML tags into the document being formatted. An HTML display system (commonly called an HTML viewer if the document is local or an HTML browser if the document can be accessed from a remote location) then parses the tags and renders the presentation of the document on a suitable display.

HTML tags come in pairs. For each "open" tag, there is a corresponding "close" tag. All tags are simple ASCII words or letters preceded by a less-than "<" character, and followed by a greater-than ">" character. A simple tag is "HTML", which tells the browser that the document to follow is marked up in HTML. This tag takes the form:

<HTML>

Since tags are not case sensitive, <html> can be used as well. This tag is the HTML open tag, and it has a corresponding close tag. In HTML, a close tag is formed by inserting a slash "/" character after the less-than character and before the tag name. Thus, </HTML> would form the close HTML tag.

Some tags require optional information. This information is included after the tag name and before the greater-than character. Such tags include IMG, which instructs the HTML display system to load and display a graphics element at the present location. Since the location and name of the graphics element is needed, it is included as an "attribute" in the tag. To display a photo called jim.gif, I would include:

<IMG SRC=jim.gif>

in my document. Notice the space between the tag name and the attribute name. That space is necessary.

IMG does indeed have a corresponding close tag, but since IMG doesn't

turn something on that must be turned off, the closing tag is seldom used. That forms the basis for using closing tags. Opening tags that "turn-on" a formatting style require closing tags. For opening tags that do not "turn-on" a formatting style, closing them off is optional. Of course, exceptions exist, but you'll rarely go wrong marking up with this rule in mind.

@(A): The BASIC HTML Tags

The following tags are considered basic since they implement either the essential or often used formatting options available in HTML. Each opening tag is listed in its HTML form, and a description of the tag is given:

Tag	Description
<html>	begins an HTML document
<head>	specifies the heading information (title, etc.)
<body>	specifies the body of the document (information)
<p>	Inserts a paragraph.
<hX>	Renders the following text in heading size X. 1 <= X <= 6. H1 is largest, while H6 is smallest
 	Line break
<title>	Specifies the title of the document
<hr>	horizontal rule (line across document)
	Emphasize text strongly (typically rendered as bold text)
	Emphasize text (typically rendered as italics)

Remember, this is but a few of the possible tags.

@(A): Creating an HTML Document

In HTML, HTML documents are referred to as "pages", and each page is constructed as a simple ASCII or ISO 8859-1 (superset of ASCII) text file. No preprocessing is necessary. This makes creating documents as easy as editing a text document. HTML files are typically given the file extension ".html", and IBM PC computers running MS-DOS typically shorten this to ".htm" due to DOS limitations. However, the former extension is most correct. Although fancy HTML generation applications exist, most people on all platforms simply create pages using a text editor. Since Commodore owners can usually find a text editor, Commodore enthusiasts can create pages just as easily as anyone. Additionally, the WWW and HTML encourage writers to create small pages, and break up large documents into linked pages of smaller sizes. Typically, HTML documents are less than 10 kilobytes in length. At that size, even an expanded VIC-20 can create full size HTML pages.

Let's create our first document. Edit a file called template.html and place the following text inside it:

```
<html>
<head>
<title>This is an HTML title</title>
</head>
<body>
<h1>This is an example of Heading 1</h1>
This is a paragraph.
<p>
This is another paragraph.
I want you to see this next sentence.  <strong>Therefore, I am strongly
emphasizing it</strong>.
Now we are back to normal.
This sentence is below the last in the source, but will appear following
it when displayed.
</body>
</html>
```

Notice which tags require closing. Also, notice how <HEAD> and <BODY> are used in the document. Notice the two final sentences in the above example. The sentences appear on different lines in the document, but HTML specifies that all carriage returns will be translated into spaces. It further specifies that if multiple spaces exist in a file, they will be reduced to a single space. Thus, using spaces as alignment helps will not work in HTML. Likewise, using linefeeds and carriage returns to specify alignment will also fail. If a new line is necessary, use <p>, which will leave a blank line, or
, which start a new line.

@(A): What's in it for Commodore Enthusiasts?

This is an interesting question, and I hope you agree with my answer. Many claim that HTML is useless to the Commodore owner since the

Commodore can't display HTML. While I am not even sure that is true, (I've heard of simple HTML viewer programs for the 128), it doesn't matter. Commodore owners who access the Internet from a "shell" account can access the World Wide Web via the "Lynx" text browser. Since the WWW is constructed of HTML pages, those Commodore owners can indeed view HTML files while online. Many Commodore enthusiasts possess useful information. Putting that information on the Internet via HTML and WWW makes it widely available to other Commodore and non-Commodore computer owners. Why worry about the latter? You'd be surprised how many former Commodore owners are coming back into the fold after viewing some Commodore HTML pages. The information on those pages triggers fond memories. Many fire off messages inquiring about purchasing a new or used CBM machine after seeing these pages.

To the naysayers, I submit that there is nothing PC-centric in the HTML standard. If an HTML viewer doesn't yet exist, it has nothing to do with the computer system. As HTML was created to allow successful operation over many different computer systems and graphics capabilities, HTML encourages usage on computer systems like the Commodore, where there are limitations in display size and resolution.

In fact, the Commodore community should embrace HTML as a markup language, for it represents a standard way to effectively mark up documentation for viewing on a variety of computer systems. Using HTML opens up a whole set of possibilities for easily created, standardized documentation publication.

Disk magazines, like _LOADSTAR_, _DRIVEN_, _VISION_, and _COMMODORE CEE_, could produce issues that contain more layout information than now offered. Since the viewer would now be standardized, these publications could possibly forego the distribution of the viewer software and offer more content in the extra space on disk. A side benefit is the ability for Commodore users to read each issue on any platform. Possibly you'll never need to read LOADSTAR 128 Quarterly on an IBM PC, but what about reading it on a 64, while your sole 128 does something else? Moving to HTML would shift a disk magazine's focus and concern from the presentation, which would become standard, to content, which is why Commodore owners read such magazine anyway. How many times has otherwise great information been presented badly in a disk magazine? Use of HTML could help alleviate that problem. Publishing a disk magazine is time consuming because not only must editors work on the articles themselves, they must also write the software that presents the articles to the viewer. Using HTML and a pre-written browser would allow editors to spend more time on laying out and editing articles.

Disk magazines aren't the only winners here. Have you ever wanted to create a small publication? The use of HTML and a third-party HTML viewer makes it easy for you to do so. Just like the editors of bigger publications, HTML allows you to concentrate on presenting your information without worrying about writing the presenter software. Now, obviously not everyone should publish their own magazine, but how about help files, information disks, software documentation, club newsletters, etc.? These publications can all benefit from this technology.

These are but a few of the benefits of switching to HTML for document layout. Other uses include upward compatible support. Using HTML allows the Commodore 128 user to view documents created for the 64 in 80 columns by 50 rows. C128D owners can take advantage of their 64kB video RAM even when viewing documents created on 16kB video RAM C128s. Publishers would no longer be constrained by lowest common denominator support. They can now include whatever they want and be assured that the presentation will look fine on all platforms. When a user upgrades his machine, he or she can immediate utilize those new features without requesting a new version of the publication. Also, for software, even though the software itself might differ by machine, the online documentation need be written only once. As well, never forget that marking up in HTML makes migrating your documents to the Internet and the WWW a snap!

@(A): Creating an HTML viewer on the Commodore

Obviously, before Commodore users can reap the benefits of HTML, we must create both a HTML generator and a viewer. The generator is easy, as HTML is simply ASCII text files. So, we are left to design and implement an HTML viewer. The following conditions should be met:

- ability to utilize all Commodore peripherals within reason
- ability to work on a stock machine
- ability to recognize and display valid HTML 3.0 or lower files

At first, we're going to concentrate on developing our viewer for the

Commodore 64, although we should strive to offer versions for the 128, C65, Plus/4, C16, B series, PET, and VIC-20. I am reasonably confident on all but the last one.

Although we intend to develop a viewer that supports the above, our initial development will operate on a much smaller scale. The first revision of this viewer will operate on the stock machine and will contain support for the basic HTML tags as outlined above. Our design will allow us to extend the capabilities to encompass our goals.

@(A): The Viewer Execution Flow

I am not very good at drawing execution flows, and the native format of this magazine doesn't lend itself well to them, anyway. Therefore, I will simply describe the execution flow.

The viewer will start by asking the user for a document to access. If the file does not exist, an error is printed and the user is asked again. If the file exists, the viewer will begin reading it. If a tag is found, the tag should be acted upon. If text is loaded, it should be displayed on the screen using the current markup controls unless the control information is incomplete. In this case, the text should be stored for later display. The file should be parsed in this way, until the end is found. Then, the system will wait for either the user to select a link or type in a new document to view.

Most of the time, text can be displayed as soon as it is received. However, there are exceptions. Some tags, like the <TABLE> tag, which creates a table on the screen, require that all the data in the table be known before the table cell information can be calculated. In cases like these, we must store the data and wait for the </table> tag.

The above flow explanation ignores some subtleties like carriage return stripping and multiple space reduction. Those are left out because at least one tag, the <PRE> tag (preformatted text) overrides those rules. <PRE> text is displayed in a monospaced font exactly as it is prepared in the document. Text is not wrapped, and spaces are not reduced. So, we will make those formatting options that are normally turned on.

@(A): Conclusion

I regret that we haven't gotten very far in the development process with this installment, but we'll make up for lost time in the next installment. One thing that I would like to encourage from readers is comments and suggestions. Do you see a problem with some of the above information? Do you have a better way to parse some of the information? Do you see limitations in the data structures? Since we haven't delved into some of these aspects yet, do you have some ideas of your own? I can guarantee that I'm ready to discuss them with you; however, I can't read your mind. I think it's important that this project be completed, as it forms the core of a successful WWW browser, and I see everyone wanting to know when one will be available. I am less concerned that my name appear on the finished product. In fact, I think a product that draws on the talent of the entire Commodore community would most likely exceed the quality a single individual can afford a piece of software. So, fire up those assemblers and put on those thinking caps.

==========

@(#)gfx: Creating 3-D Dungeon Crawls
         by Todd S. Elliott (telliott@ubmail.ubalt.edu)
         http://ubmail.ubalt.edu/~telliott/commodore.html

@(A): Introduction

What? Another article in C=Hacking that deals with the subject of 3-D? Well, not in the same vein as Mr. Judd's study in 3 Dimensions for rendering shaded 3-D polygons in real time. (See Polygonomy in C=Hacking 12.) No, this article only deals with the aspect of the 3-D look and feel in those dungeon crawls you see for the c64. Some titles spring to mind, like the gold box series by SSI in collaboration with TSR, i.e., Pool of Radiance, Curse of the Azure Bonds, or other popular titles such as the Bard's Tale Trilogy.

With the techniques described, the aspiring Dungeon Master (DM) can create a rich world to torture his players at the local terminal of the beloved c64! That, and some generous helpings from the local pizza delivery company. "Hey! Look out for the grease! Arrrgh! Now the `A' key is stained!" ;)

@(A): Nuts and Bolts

Let's begin with the 3-D screen. It is comprised of a 12x12 square of custom characters, which never change position. The 12x12 square looks like this:

```
characters 01,02,24,25,48,49,..,96,97,120,121
           03,04,26,27,50,51,..,98,99,122,123
           ..
           ..
           22,23,46,47,70,71,..,118,119,142,143
```

The 144 characters are positioned in an unusual way: they flow in two character columns, run down for 24 characters, then go back up for the next two-character columns. Think of these two-character columns as SEAMS in the 3-D window. Right now, there are six such SEAMS in the 3-D window for the dungeon. Of course, these are not the actual characters (screen codes), (I forget what they are right now), but they are in a continuous sequence, i.e., no broken or interrupted series of screen codes. (If memory serves me correctly, they are the last 144 screen codes in the 256 screen code table.) The corresponding color codes never change, for the sake of speed.

Next, we deal with the concept of CELLS in the 3-D window. There are a total of 13 CELLS which we can utilize individually to show an object, which in turn, is displayed in the 3-D window in the correct perspective. By objects, I mean walls or doors. The perspective is from the user's standpoint. This creates the illusion of the 3-D look and feel, but does not simulate true 3-D rendering on the fly such as Polygonamy by Mr. Judd. (See Polygonomy, C=Hacking 12.) Let's take a look at all 13 cells, to give us an idea of what each one does:

```
Cell 01 - Farthest left side object.
Cell 02 - Middle left side object.
Cell 03 - Immediate left side object.
Cell 04 - Farthest right side object.
Cell 05 - Middle right side object.
Cell 06 - Immediate right side object.
Cell 07 - Farthest front object.
Cell 08 - Middle front object.
Cell 09 - Immediate front object . (Currently used for backdrop only;
          fills the entire 12x12 screen.)
Cell 10 - Farthest left side object situated in front.
Cell 11 - Middle left side object situated in front.
Cell 12 - Farthest right side object situated in front.
Cell 13 - Middle right side object situated in front.
```

The 3-D engine, before it starts to redraw the 12x12 screen, checks the user's (you!) orientation. If you are facing north, the engine will know this and configure the 3-D window accordingly. Let's assume that the user is facing north, and the 3-D engine now looks in the dungeon map for relevant objects to place on the screen. The 3-D engine will look for doors or walls only. In future revisions, this is expected to change. Currently, the map value for a wall is 128 and for a door is 7.

First of all, the 3-D engine looks in Cell 3. If it finds an object there, it will paint a wall or door and will skip the search in Cell 11. The reason why Cell 11 was skipped is because an object was found in Cell 3, which would overwrite Cell 11. We don't want the 3-D engine accidentally overwriting Cell 3 with Cell 11 on top. Next, it searches for an object in Cell 6, and if it finds an object there, it will skip Cell 13. Last, it will search in Cell 8, and if an object is found, it will skip all remaining cells except for Cells 10 & 12. This is to ensure that there are no overlapping cells which result in a less-than harmonious 3-D look and feel. This hunt and eliminate approach employed by the 3-D engine can be referred to as a first-last approach. There are three layers of information for the 3-D engine to process, and it starts from the 1st layer to the 3rd layer, minimizing conflicts and results in a natural 3-D look.

Here's the sample code for the direction of north:

```
   ;paint the north surroundings
   ; Note: the .Y register refers to the location in the map for the 3-D
   ; engine to search.
   ; position the paint location
   npaint  =*
           lda #101
           sta subtract+1
           sta addition+1
           jsr minus
   ;first block module
```

```
          ldy #100:jsr cell3
   ;second block module
          bne +:ldy #74:jsr cell11
   +      ldy #102:jsr cell6
          bne +:ldy #78:jsr cell13
   ;third block module
   +      ldy #76:jsr cell8:bne +
   ;fourth block module
          ldy #50:jsr cell2
   ;fifth block module
          ldy #52:jsr cell5
   ;sixth block module
          ldy #26:jsr cell7:bne +
   ;seventh block module
          ldy #0:jsr cell1
   ;eighth block module
          ldy #2:jsr cell4
   +      ldy #24:jsr cell10
          ldy #28:jsr cell12
   ;position the party
          jmp plus
```

@(A): Drawing the Screen

Now, on to the actual drawing of the 12x12 3-D screen! First, the 3-D engine immediately draws a backdrop to Cell 9. This is the floor and the sky you see in the 3-D world. (This step may be unnecessary in the future.) Then, the 3-D engine takes the object found in a particular cell and draws the object on the SEAM in the 12x12 window. Remember the SEAM's, eh? Depending on the size of the object, the 3-D engine may encompass two or more SEAM's in one sitting. First, it takes the pointer values from the graphic tables, extracts the raw graphics data, and stashes the same raw data on to the character dot data area. Please note that the 3-D engine does not stash the data to the screen; only to the character dot data area. Remember that the 12x12 had a character grid- the VIC-II chip continuously updates the characters with its corresponding dot data. Hence the reason why the characters never change in the 12x12 3-D window. This is needed for two reasons: One, the 12x12 grid uses only 144 of the 256 available characters, leaving some left over for regular character. Two, it allows the 3-D engine to `unroll' the graphics updating loop using self-modifying code , resulting in speed increases.

Here's a sample code for the grunt screen updating routine:

```
   ;to paint the 3d surroundings
   paint   =*
           lda #59:ldy #128; This is the lo-hi byte representation of
                           ; the character
           sta dummy+2:sty dummy+1; dot data area.
           lda <milleu:ldy >milleu; the pointer to where the backdrop
                                  ; can be found.
           sta dumb+1:sty dumb+2
           ldx #$03
   -       lda #$00
           sta disflag,x; this flag is used for hidden objects.
           tay
   dumb    lda $ffff,y
   dummy   sta $ffff,y; This is the self-modifying code to draw the
                      ; backdrop.
           dey
           bne dumb
           inc dummy+2
           inc dumb+2
           dex
           bpl -
           ldy #127
   -       lda 22016,y
           sta 16256,y; The remaining part of the backdrop is drawn.
           dey
           bpl -
           jmp direction

   ; routine for printing two char wide column on the dungeon window
   table   =*
           lda cassette,y; retrieves the pointer values from a table.
           sta twain+1; The table is stored in the cassette buffer at
                      ; 820.
           iny
           lda cassette,y
           sta twain+2
```

```
            lda chartable,x
            sta seam+1; This retrieves the pointer values from a table
                      ; for
            inx       ; the character dot data area.
            lda chartable,x
            sta seam+2
            ldy #192; to output enough bytes to fill 24 characters.
            twain lda $ffff,y; Self-modifying code used here to draw the
                              ; 3-D screen.
   seam     sta $ffff,y
            dey
            bne twain
            dey
            rts
```

@(A): Conclusions

Whew! The 3-D engine has finally done its work and waits for the user to press a key for a new facing. The 3-D engine by itself is quite small and flexible enough to handle as much as the programmer wants to throw at it! The power is in the tables and the 3-D hunt/eliminate routines.

The 3-D Dungeon Demo can be found in this issue of Commodore Hacking (Reference: code, SubRef: gfxcode), on the Commodore Hacking MAILSERV server (Reference: code), at http://www.msen.com/~brain/pub/dungeon.sda or in the file DUNGEON.SDA available at my site. There may be a c128 version in the offing, but in 40 col. mode. Of course, there are no planned versions for the c65. ;) Please note that it does not contain the source code. However, upon request, I will be happy to send you the source code in Buddy format. (Right now, I'm trying to make the source code assembler neutral to work in either ACEsembler or the Buddy assembler.

Right now, I have already done work in producing a Dungeon Master's environment- with a 12x12 screen grabber routine and a Retouch Studio. The 3-D engine will be overhauled completely to create a 3-D environment in the hi-res multi-color screen, as opposed to using custom characters. In the future, I hope to have a complete environment, where the user can design dungeons, comment them, add a bestiary, add custom doors and walls, and map editors for the purpose of playing pen & paper dungeon games. This way, the program only shows the visual aspects of the pen & paper genre; it will not have combat or character interaction. I expect a version to be ready by the end of summer '96. I'm not sure how I will release the software, but I will choose an appropriate medium for mass distribution that is accessible to C= users.

That's it! Feel free to drop me a line regarding this article. I'd be happy and will try my best to answer any questions or comments about this article. Until then, Happy 8-Bit computing!

==========

@(#)trivia: Commodore Trivia
by Jim Brain (j.brain@ieee.org)

@(A): Introduction

As some may know, these questions are part of a contest held each month on the Internet, in which the winner receives a donated prize. I encourage those who can received the newest editions of trivia to enter the contest.

This article contains the questions and answers for trivia editions #27-28, with questions for edition #29 and the current contest, #30. Why two sets of questions? Well, as some may know. I have recently moved, and that has put me behind in posting answers. At present, my reference books are still packed in storage, so I can't finish the answers.

If you wish, you can subscribe to the trivia mailing list and receive the newest editions of the trivia via Internet email. To add your name to the list, please mail a message:

```
To: brain@mail.msen.com
Subject: MAILSERV
Body:
subscribe trivia Firstname Lastname
help
quit
```

@(A): Trivia Questions

Q $1A0) Commodore produced an assembler for the 128 called HCD65. What does HCD stand for?

A $1A0) Hedly C. Davis, the writer of the assembler.

Q $1A1) Who wrote most of RAM DOS?

A $1A1) Although many assume Fred Bowen wrote RAMDOS, Hedly Davis actually wrote the bulk of it.

Q $1A2) What is the name of the first C64 disk copy program? (hint: it sported a "gas gauge".)

A $1A2) 1541 Backup.

Q $1A3) What was the case color of the original Commodore 64s?

A $1A3) Ivory, just like the case color of the VIC-20. In fact, early cases WERE VIC-20 cases.

Q $1A4) There are at least two ways to enter 64 mode from 128 mode on a C128: go 64 and sys 65357. They produce the same result (64 mode), but they differ in at least one noticeable way. How?

A $1A4) sys 65357 doesn't ask the "Are You Sure?" question.

Q $1A5) What CPU powers the B-128 computer system?

A $1A5) The 6509 CPU.

Q $1A6) What type of drive mechanisms are in the D series hard drives from Commodore?

A $1A6) The D9060 and D9090 drives used "Winchester" hard drive mechanisms.

Q $1A7) Commodore produced a 16kB RAM expander for the Commodore VIC-20. What is its model number?

A $1A7) The VIC-1111.

Q $1A8) Commodore produced at least one disk drive with an optical track one sensor. Which drive?

A $1A8) Certain early versions of the 1541C drive had a functional track 1 sensor. Later, due to compatibility problems, it was disabled, and then later, the sensor was removed from the mechanism. In addition, 1571 drives and 1581 units have optical track sensors.

Q $1A9) The Commodore PET series used the IEEE bus to communicate with peripherals. Each peripheral had a unique ID. What range of IDs are supported by the PET?

A $1A9) IDs 4-15 are supported, although you cannot connect all 12 devices up at one time.

Q $1AA) Many people have developed Commodore software with the PAL assembler. What does PAL stand for?

A $1AA) Personal Assembly Language (PAL).

Q $1AB) Many people remember Compute's Gazette. This magazine is best known for the word processor program it shared with thousands of subscribers. Name the program?

A $1AB) SpeedScript.

Q $1AC) In some 6502 assemblers, the opcode "bge" is available. It stands for "branch if greater than or equal to". What more common opcode is this opcode referring to?

A $1AC) bcs (Branch Carry Set)

Q $1AD) If I wanted to do a "blt" (branch if result less than), what 6502 opcode would I use?

A $1AD) bcc (Branch Carry Clear)

Q $1AE) Each Commodore peripheral has a device number, which is associated with a type of device. 8-15 implied disk drive, 4-5 implies printer. These have remained constant from the PET to the C128. However, one peripheral in the PET was phased out and its device number was reused. What device number was reused?

A $1AE) Device #2. The PET systems used #2 as a second tape drive, but in the newer computers, #2 refers to the RS-232 port.

Q $1AF) What is the maximum amount of general purpose RAM can one utilize in a stock C64? (I need an exact number here)

A $1AF) In the Ultimax memory configuration, if you guarantee no interrupts can occur, one can utilize all but the first two memory locations for general purpose RAM, giving 65534 bytes of RAM. If you can't guarantee you'll never receive an NMI, you lose 2 more bytes for that vector, giving 65532 bytes available.

Q $1B0) What was COMPUTE!'s original sub title?

A $1B0) "The Journal for Progressive Computing".

Q $1B1) After COMPUTE! was absorbed by General Media, how did the name change?

A $1B0) The name, having gained an exclamation point and lost a period many years before, reverted back to the period as the ending punctuation.

Q $1B2) What Commodore content magazine was named after a nautical term?

A $1B0) "Ahoy!"

Q $1B3) What Commodore content magazine was named after a BASIC keyword?

A $1B0) "RUN"

Q $1B4) What CPU gets control first when a Commodore 128 is booted?

A $1B0) The Z80 CPU has control first.

Q $1B5) What CPU powered the Commodore C900?

A $1B0) The Zilog Z8000, from the company who brought us the popular Z80.

Q $1B6) How large is the monitor installed in the SX64?

A $1B0) 5" diagonal.

Q $1B7) What color scheme does the SX64 boot up into?

A $1B0) White screen with cyan border and blue text.

Q $1B8) What is printed as the stock SX64 boot up screen?

A $1B0)
```
     *****  SX-64 BASIC V2.0  *****
 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
_
```

Q $1B9) The SX64 has a reset switch behind the door that holds the monitor controls. What is strange about the rest switch?

A $1B0) The reset switch only resets the disk drive. Most people assume it resets the entire computer system.

Q $1BA) What common port is not included on the SX64?

A $1B0) The Cassette Port.

Q $1BB) In the mid 1980's, a company called Berkeley Softworks created a graphical user environment for the Commodore 64. What was it called?

A $1B0) Graphical Environment Operating System (GEOS).

Q $1BC) Berkeley Softworks eventually changed their name to what?

A $1B0) GEOWorks. They now develop the GEOS OS for Personal Digital Assistants (PDA).

Q $1BD) Most everyone is familiar with MSD disk drives. What does MSD stand for?

A $1B0) Micro Systems Development, Inc.

Q $1BE) On the NMOS 6502, what two addressing modes have but one opcode

each that can operate in that mode?

A $1B0) Actually, there is only one such mode, indirect. jmp (xxxx) is the only opcode that can utilize that addressing mode.

Q $1BF) How many transfer register opcodes are there on the NMOS 6502?

A $1B0) 6 (TAX, TAY, TSX, TXA, TXS, TYA).

Q $1C0) What are the two configurations for the LORAM, HIRAM, GAME, and EXROM pins that will allow the use of a full 64kB of RAM in the C64?

Q $1C1) What is the first thing that the C64 (and VIC) KERNAL does upon powerup?

Q $1C2) What KERNAL routine is used to set a DOS channel to input?

Q $1C3) What KERNAL routine is used to set a DOS channel to output?

Q $1C4) Before calling the routines in $1C2 and $1C3, what register must you load?

Q $1C5) What 3 devices can the KERNAL NOT load from?

Q $1C6) In the Commodore KERNAL, there are "high" and "low" level routines. To which class of routines does "SECOND" belong?

Q $1C7) If a programmer calls the KERNAL routine "STOP" and the RUN/STOP key is NOT pressed, what is returned in the .A register?

Q $1C8) The Commodore KERNAL routines are all accessed via a jump table. What routine is used to change the values in the KERNAL jump table?

Q $1C9) A call is made to a KERNAL routine, the call returns with the C bit set and the .A register holds $02. What error does this indicate?

Q $1CA) If a call to READST is made, and a $40 is returned in .A, what does this indicate?

Q $1CB) What routine can be called to determine the physical format of the Commodore 64 screen in characters?

Q $1CC) The Commodore 64 starts a non-destructive RAM test at what location?

Q $1CD) Which way does the RAM test proceed: up or down?

Q $1CE) Which KERNAL routine is used ONLY in conjunction with a Commodore IEEE card?

Q $1CF) Many hybrid BASIC/ML programs use SYS to transfer control from BASIC to ML. However, a few use USR(X). When using the latter function, where does BASIC fetch the ML routine's starting address from?

Q $1D0) To load a program from the current location on a cassette tape, what two key combination must a user press on a VIC-20 or C64.

Q $1D1) If I issue the BASIC statement OPEN "JIM,S,W", What type of file am I opening?

Q $1D2) Is BASIC in the Commodore computer systems an "interpreted" or "compiled" language

Q $1D3) What type of variable is A%?

Q $1D4) If I issue the BASIC line PRINT:PRINT "A","B" what column does the "B" show up on when run on a C64?

Q $1D5) What column does "B" show up on if I run the BASIC line in $1D4 on a VIC-20?

Q $1D6) Alphabetically, what is the first BASIC 2.0 to have a 3 letter abbreviation?

Q $1D7) How many times does the statement FOR T=1TO0 execute?

Q $1D8) What base does the BASIC LOG command use for its logarithm function?

Q $1D9) A = NOT B can be written as which expression:

a) A = -B
b) A = -(B+1)

Q $1DA) What does INT(-15.43) return?

Q $1DB) What does ASC$("JIM") return?

Q $1DC) What is the abbreviation for GET#?

Q $1DD) What is the largest integer value that Commodore BASIC can handle?

Q $1DE) What is the ONLY Commodore Editor key not affected by "quote mode"

Q $1DF) What is the range of RND?

==========

@(#)error: ? DS, DS$: rem The Error Channel

We here at Hacking Headquarters are actually perfect, but the online networks occasionally globally alter the issue after it leaves our hands. The printed version is similarly changed by the print shop. We think it's an attempt to discredit us, but we will thwart them again by simply printing the corrections to the mistakes THEY introduced below.

@(e)dbldma: Speed up RAMLink transfers with the Double-DMA Technique

Brett Tabke wrote in to correct some misinformation contained in this article in Issue #11. In the article's discussion of RAMDOS, it was written that "RAMDOS continually pages its code in and out of main memory" during transfers. Mr. Tabke, who has researched the RAMDOS code extensively, notes that the above is incorrect. RAMDOS pages the main code in to initiate the transfer, but the bulk of transfers are handled by the 256 byte interface that remains in memory at all times.

@(e)mags: Hacking the Mags

We reprinted Jeff Jones' electronic mail address as printed in LOADSTAR LETTER #31, but Jeff sent us a note mentioning the address had changed, and the correct email address is jeff@loadstar.com.

@(e)cmdcpu: Underneath the Hood of the SuperCPU

In this article in version 1.0 of Issue #12, there were two references to "Exploiting the 65C816S CPU", an article pulled from the issue for space reasons. We regret the error. The full article on the 65C816S appears in this issue (Reference: cpu).

@(e)gfx: Taking to TED: The MOS 7360/8360 Display ICs

In early versions of Issue #12, the TED IC article was incorrectly attributed to Harsfalvi Levente. Hungarians customarily sign names with the last name first, opposite English notation (implying that the other way must be Hungarian notation :-). The article should be attributed to Levente Harsfalvi. Version 1.3 of the issue fixes this problem.

@(e)polygon: Polygonamy: A Study in 3 Dimensions

After the publication of this article in Issue 12, Stephen Judd noted that the following information was not included in the article:

Memory map:

```
$0800-$1BFF  Tables
$1C00-$1FFF  Color info for bitmap #1
$2000-$3FFF  Bitmap 1
$4000-$58FF  Fill routine for bitmap #1
$5900-$5BFF  Tables
$5C00-$5FFF  Color info for bitmap #2
$6000-$7FFF  Bitmap #2
$8000-$A6FF  Code
$A700-$BFFF  Fill routine for bitmap #2
$C000-$C5FF  Yet more tables
$C600-$C6FF  List of points for plotting routine
$C700-$C7FF  Fill patterns
$C800-$CDFF  A few more tables
```

The fill pattern table may be broken down further. Each fill pattern is eight bytes, so to get the address in the fill table multiply the pattern number by eight:

```
        0 - Empty (clear)
        1 - $FF (solid)
        2 - Brick
        3 - CrossSmall
        4 - Inverse of 3
        5 - Dither 1
        6 - Dither 2 (inverse of 5)
        7 - Zigs
        8 - Zags
        9 - Zigzag
        10- Holes
        11- Smiley
        12-15 Not used
        16-23 Shockwave    \ Animated patterns, eight frames each
        24-31 Squaredance  /
```

If you have a freezer cartridge you might want to try changing the patterns. You might also turn on the multicolor bit (bit 4 of $D016) to see what a multicolor Polygonamy might look like, and change the patterns (not to mention the color info) to be more multicolor-friendly.

@(e)trivia: Commodore Trivia

In early versions of Issue 12, question $186 in the Commodore Trivia article was incorrect. The correct answer appears below:

Q $186) What is the maximum size of RAM available for use for program storage on an expanded VIC-20

A $186) If you discount the screen area (512 bytes) and Color RAM (512 bytes), up to 28159 bytes can used for BASIC programs and variables (original 3583 bytes and 3 banks of 8192 bytes each), and up to 40448 bytes can be used for ML programs. (0-32767 minus 512 bytes for screen and 40960-49151).

==========

@(#)next: The Next Hack

Hacking Headquarters>look

You are in a room where editors and authors are busily preparing issue #14. You scan along the room and note the following paragraphs hastily left on the various desks:

- In part 2 of "Using HTML on the Commodore", Jim Brain will delve more deeply into HTML parsing engine design and relate some of the tricky cases the parsere engine will have to deal with.

- In another part 2, Alan Jones continues on in his discussion of complex computations on Commodore systems. In this installment, Alan will jump into Linear Programming. Equations and algorithms will be presented, as well as sample code implementations.

- Attention all VIC-20 enthusiasts! If you need some good technical information on your beloved machine, look no further than our next issue. Pinout diagrams, cartridge tricks, and important memory locations will be presented.

- In a new column, "Twiddle the Bits", Todd Elliott will show how to modify a C128D to include a CMD HD and a FD4000. Todd will detail how he created his "Tower of Power".

- Although we didn't have room this time, Commodore Hacking will begin reviewing new software titles, starting with the recently introduced "Novaterm 9.6", "The Compleat Lee O 128", and "The Compleat Crossword".

- And, of course, C=Hacking's regular goodies.

So, fire up that label program and print off one for your copy of Commodore Hacking #14.

==========

@(#)code: Hacking the Code

Being a technical, developer oriented magazine, some articles featured in C=H include executables or other binary files as part of the article. All such binary files are included on the soft copy of this issue in this

section. In an effort to retain the integrity of such binary files through distribution over various computer networks, the binaries in this section have been encoded using the UUcode format, a popular Internet binary-to-readable text encoding method. In order to execute or otherwise utilize these binary files, one must feed this section of the magazine to a UUdecoding application. Typical examples include UUXFER for the 64, uudecode on the ACE OS for the 64 and 128, and uudecode on most UNIX OS machines. Some encoders can decode multiple files, while others will require the user to manually split this section into individual pieces prior to decoding.

In addition to this section, there are other ways to retrieve the binary files featured in this issue. For those with World Wide Web access, the files are available at http://www.msen.com/~brain/pub/ To retrieve "dim4.lnx", simply access the URL:

http://www.msen.com/~brain/pub/dim4.lnx

For those with electronic mail access only, the Commodore Hacking MAILSERV server also contains a copy of these files. To retrieve a copy of "dim4.lnx", send the following email message:

To: brain@mail.msen.com
Subject: MAILSERV
Body of Message:

send dim4.lnx
help
quit

For some articles published in Commodore, the author or authors may also have other methods for accessing files mentioned in the article. These methods are described in the respective article.

Commodore Hacking always attempts to provide the reader with as many options as possible to retrieve uncorrupted binary files. Although none of these above methods is foolproof, the added redundancy helps overcome any shortcomings.

WARNING: The UUCode format translates files from binary to ASCII, not PETSCII. Therefore, either decode this section before downloading this section to a PETSCII mode computer system, or download this section without translation to PETSCII. Some decoder programs can handle PETSCII converted UUCode files, but the practice is not recommended because conversion is typically done in a telecommunications program and accuracy in translation cannot be guaranteed.

@(A)democode: Binary for Dim4

```
begin 600 dim4.lnx
M`0A;"`H`ES4S,C@P+#`ZES4S,C@Q+#`ZES8T-BS"*#$V,BDZF2*3$1$1$1$1
M$1$B.IDB("`@("!54T4@3%E.6"!43R!$25-33TQ612!42$E3($9)3$4B.HDQ
M,````T@,B`@*DQ93E@@6%9)22`@(%=)3$P@0T]23$59#2`X(`U$24TTH*"@
MH*"@H*"@H*"@#2`Q-R`-4`T@,S0@#41)331254Y-12U214%$344-(#4@#5`-
M(#$P,2`-1$E--"Y.04U%4Z"@H*"@H`T@,2`-4PT@,S@@#4U!24XT+E.@H*"@
MH*"@H*`-(#<T(`U0#2`Q,#`@#4=205!(24-3+E.@H*"@H*`-(#4P(`U0#2`V
M,B`-3T)*14-44RY3H*"@H*"@H`T@,3`Q(`U0#2`Q-3,@#4U54TE#+E.@H*"@
MH*"@H*`-(#8P(`U0#2`S,R`-1$E--"Y415A4H*"@H*"@H`T@-S0@#5`-(#(W
M(`T````````````````````````````````````````````````````````
M`````````````````````````````````````````````````````````````
M`````````````````````````````````````````````````````````````
M`````````````````$(#`C)!Y[_K#8U-@"@`'CN,-!.$=#F`;D*%IE``\C0
M]TQ``Q8\"Z(%TX)JP;4@6UN@VUL@7)S<G`#IZNSN\/+T]OC\_@`#!PD-#Q,5
M&1TA)2DM,S<]0TE/5UUE;W>!BSD`M'$JT'PAVN'X,_//M*D'9'\'`QH/)+0#
M:[!#4Y5!X`R`:$5EU,"W_4:0R(6PJ'%TU:H4//,?SNHQ4_"%2=(OTR&D'2(Y
M=28B\2DM%4G+E)&CM4$_8B:9@`^(0MP=D&)6LZOYGU_5-55PI<!<E;=404/M
M2'(!'@H'"L(^$8(8B*C@E),([%6_TEY`Y%:H&C?',N8R)4$'CR$@4Q-F[@A&
M^RI",9,)L!Q+!R^:V/M!$4*/<+M)_Q&*"OLGKMB-$TVK$!V9`X:PH3*`IS)J
M2A5AO_@4RP"W0@CH;]*`!T!G#"@#=B@#.13!/?2%*(IBRH$`YL%7!%#-!;Z)
M#ABL#-&>J@\R.T'`1#R(0[MB=.BPFVOQM8;-&LCG?^GRL!?:",W^5D(03`KZ
M#,T'F^I0[O$O#)0$=$/,_UB)YIND"MW02L@L)94VDP(*987\4S8(;^.4(54R
MS]`1)QNB-+1M.I94U``U4F"[=-^<-`BN:"--A<A8N(U$%2)DX3%258B<<;O[
MK%+^8IN54L$W&0MJI<LM1S6U(!(613RD8_C[=+G2[(!GD)P)F4]'42UT!>F?
M2+$@`&*K>8$T/4YZAC`EC'-A"'*%H14BR/PUOQ1QLEKNODBV/&#SF2U&X)T1
MF)@RWY"\ILVT+$4\*XU&O2DKMX*'UAR+@H4RT^",/=NZ$$YC3W=K$<LP0I)0
M7ZQ2UNNTT"H)`9F#?*Z)N<9HLXWPUZHYGH1:?[*ECLU>(AL_$,$L2$F0A9R;
MV%2$;RI*R"BE0E2A(>`M5..R)`_`&`B%,=K>C7-;'YX\)'8%R`]%JX`HJ8S#
M*.,0BA)?`2:Q".$I?6LIXO18):.C5")+1<A4W"WK(>YZTRFI=4-J!F(Y,:"D
MS_;DJ(;4,3#0@8@25!K!4U4"$^\-U)$$`!IG[(Y+$6"Q7R&T$T*@%%6`V#&Z
M<=S&!5\"+!""*^O5S^(TR$2LK%;V"B.N2QA9;!9._!88TH`*Y$C*$4%,@DC_
MIPZ^,01:5)!H"1$D5)"002=`4'3H^)@'3;D\0W)1L7!5R`Z@!4S=%X*<"B"*
```

MH&8F`CDP((HDK=6#155A](X+HF"2BA&]D=+[98F5T*H15=;2S#2;PE:P82`J
ME?6`H%J5@6C3,CP0=[O=[J(GH(-?_9)AX(/B)"K"**L;*G78ZH9*(;:ZH5*)
MK6XH4\KJAIJJ$V0,8&-"C3%`C`DZQ@`L)GR:5#?(!!)#6)VPJ>(R!G@QP<48
M0,6$&6.`$A-PC$%[4\755-W`INI$(@58,$J%L&@=I4)XM)!2(4Q:2:D0+BVE
MM+B,1'76-&1EB&)!:SE@C`%L3*@Q!H@Q0<<8@,6$SQ.&`"<FD!C"G`G")QPI
MQ@`O)K@8`ZB8,&,,4&("CC%H?W+1>P0YF^6<^.!,;!]$&('"/56`@9ZD"C;+
M08*@@8D61%`H$B3DH5ES`]4<1@NG#L\C*F!TA?B,3/O<`*G4+X"15B#02FP5
M&B)_2U[.=)1$#!;>F%2HZZ((RDJ0`AF*F`G$+B=8'/8G!3$1A!U9]EM7[21M
M$D;4\,J;_9GI?A3>"@;O$$V-S@N17/9!)L17FF:]@,K!,H$X*4X25LX/02"(
M2@LH1"L>?FZ`J>JB=+CP`JFJ24/!8D.X$M>XD1>L54KA`)):J@'(E<RTN)_5
MC=+P$_$%21.AU=>PFC(A@&63.$+J?W"U_J6A&/6H;@G*T4*C(TJ*N>PL-.XN
MZKYU1Y3Z1^G,_ML&KA<`.8CZHF)A/[$_+(]@T0.06P0",&W"9R3W2CA(F"6Q
M@H_K$-`0R\\@%00J6R\-CBA&YL'::$:I.-#O2&;]I5%E`W$>[3`.*1H.K/<`
MM@AP8LR`RODMB());EF_C[@/$33?TJBCQADFGVLA2?"^:3@I?^B/%)>+A$-#
M+EL<?)Q\?#]?Z])0$MT*T^=_?#-/7..JVV&L%"XY&)9`H$F@E\$$4V'2()BS
MN=WM4C-R-'EJ=KMHI&(-P9[:>OOJ:@2ZN1ET9.:`8^AYL,*!BJD:JU:R+V@1
M.H82MT5XMPS$.<:C#B'6(BWR[F@W4(`,[5_54FP7K!16,,0^(S8@0TPWD@6@
MAZ35:N3PZC!"Y3,0Y7-@Z>[FG2A"YX-PPW%5M:N0=^><D1L.BM1@IJ21$F<B
M56021VXD&SN+8WHEPV0DPUQT:CUS$9A6QH4HZ#/T$(#%^35\0B*9>5GJP("2
MS107T`TQ_YM4H`$1IY(Q,J%5*O!@G[CJ^JH39NZO.C'W77_?+[C<90JZ)\(<
M0O2ES25D7^*<0@`U"^DW_CQ7T*]S!OV#?OM&_9IYVYQ"=:$:YQB2#<7TB;&%
M2(]GM)I2(K6:L:1(?CV+B\RM\E_(2/R:T4U1R9(&"`F*D->-DMQ[I;K_1%),
MHG:,D)P(U]?A('(N.CF<4@\WNG@470@29D8G[%2K*F3$==9'9R!20[57:$YT
MUIR(;Q;57![?AGOHGZ/JZZ[ONKO^O^*#@/=2TW$1L:X>@>P-NER;O71/A$](
MNR<PB.72ZHGNB78AB"M`",G%N<,#1(30+:(,&]&)&M)I1^)%@R/B)2.1"W06
M`3`Y`3`(`3`.`3`D`3`&J?:=!P,`QU%5ID("G=Y&`)A/\0E0+"1``2)$?10Q
M`V^C(,:BP`,I\%I#2L:`CA!M$!056HF+6ND!6!R@-F!`P%$0(&`(&(13Y:[C
M(W+7\Y-XCW<?YA[O.DP]#A3?.!0!W?&-0R&0WK6[>`QWYW@2-$>#B&Y\QW=\
M$;IC.\;CF=`<3X7FR+#0$%,EZA`%*>D!(R22(*(B&J$Q;V0C5"(*4E++"RF)
M(BKB$5K1,?(A0("N3G$FO%[F:1.D6=K4:)HF\0HC11E2\3L_!!N#U@<-NH9Q
MI,BX']614,>%@H=^!]D98H^1&Z(PP`9V1@).AY4'K1`\,"Q".#B0%+L=&^
MH!D@`S`[607W_'6G,9*T*=6D9)L2>O*6.!(GV9A;'+.FJ5;E*%P#94G]9W+#
M*#IZ:HJ:,L#D47'QT14?/S='3_'4W/STU=<?'34W/ST%!PT/$1,4'#$S-#P
M:`#B.]Y%'``0$:&X4'P#NN,]WF,O8)#8C@?%=:`\G@3-$:$@(>)IT!P5#A8B
M_@`<B7]`1^(NT!_7@?[X$VPD[@7UL<,@??P/VN./D)&HD;`1T4/A(^*ZT*^?
M@2/Q;^A(W!WZX_K0'W^(C<0]HCYVY'/C/]$>,?(918V*C1@ZBX]$CFA'\1W?
M$8)A(B)%\:+X-G3'>[S'WD$BL1TGBNM#>3Q7WD@141'QE&B.%A,7$?_Q']],
M-_[C/YK4<V,_HA1?D>Y8CRG%=:0]GB3-T:3D).*)TAQ54E9B.D59P.#$!-I"
M)F<F4!>T.#6!OK#-N0D4)HPN5,64V9T)5":M+F3-M-V]"94&3FXN;J@U8F9G
M8T.U`4-+"QOJ#9W=7=U0<=3<WM:&FD/&UE8V"$UT(78F6Q1UHDX\(HHJD3&5
M96PJ9>OM9FBJI32<TKLH1:^$1^<@XI23]U6(TNX\=I-.(6H1T4*T(R*&5A(/
MX?-.V%KO_"A$"'J4_"6Z>A7A)%=FW.67@1"NU,B1Z#)HL!@&"FH,(B2C@`"`
MJ+1Z8"SZZ1NPSIL_XKL88#M:?H:>[46F#G?O-Q3H(VI1%0VH51!JS`@U)3".
M"LU>FE`X(!`O38VY)L,AD_&0J8;B[H:J"!"39!Q$,A`BQ5!>*"]A4*OE(H!J
M+N!P7%RBA50N+55HQ*!&)S>`R00M@"<%<90-&,=`CCNBM5$'IA`2FBUIP6Q9
M4V:%IDB9L$0`2H+'2$XG`E0B#LF&4><5N%J0)4A-JQ2Y7&MJ``X!WS58&>>5
MM%<29*7-*UEH%11-5@3GE;17$F6ES2MA("F`P!ATD@R0BY!2BG^4H*N-(,*E
MPDP(B&U`QL0H&V)A'E+`'$%TH$P_'E%;K$,$)#204$%(9\00B)1!A4%08<35
M(2(2FDBH(J2+$$,BN5I4D6-#&RE"10]2`2I"2/96RTX,866(>'0FHX_"`G1N
MI@3U*-(,42+E01P5*>]$0:J3L)7#4B1V8C5V4N:Z*)B4;#J;TI!0X#G&UD(H
MUWVYI$13R9"Z0#RY]YFY-\,2S/Y:,$HB7^6F6LLOZ">),A0"45:M5\10>M9*
MK??-">5:IWCGW*250@FQU$Q)"ZE=I+U'@Y.D*Z*^DA*KZ?B"6))?LC8EM2V%
MB(8P]MX4(I)LSBD1&$&'T+@'@8@G3XMG8AIL-V3J]A&^I%3@"W0/R46Q\TZ:
M1M6LP,2HF3QOM,D6&K>6-_L2<%9:`@]J!O%!E&>14'H+3Z'@@Y!Y):Z@&&FI
M0=:AMRL5V[.P#=#0$+%5'L:E0XN*>'*P0T+=H:9/^]`0`=7$SH?@M'1H45/.
M4RSL?&N:-^V9AH:()J(<A$N'%BD%+85]$/XO0U'^T**F7O6'%BU5DD-KCE)*
M<NBHBA0EJHFL+"Y#,QE:]-/>9!KBR?&)DOT4AFX2N13FLS>C44(,X_$!BIVL
M-2(9)282X5H8`HK,$F4A5A@OGFZE6U5!A%#L'(&J(U$X(VB.!?'Y"0I$$%08
M1,-DA=4O*2RC/,FAT:>YTU-N&"J2GKPBS(@GH_1,6IR0VCDA5#0P-1%?\H$*
MAH(+`RV2XDJQ,`DC`T!(,T2=&QP>H>DF:BXTCYH,2$,Q&X)\`AL81NE1HD!&
M()Z;`3```!*Y`!>9(P?(T/?&`2D%C2#0Y@&((!P$\`NB""#*`R`-!(C0]:`?
M(!P$H@2(A/OP,\K``K`&RB#*`Q`&(,H#&&D$J+EC!ZH@R@,8>7`'JJ7]>7T'
MA?W0#\``$D`O@$-``J0S0WB#*`XH8;0X$JJ7];0\$J(HXY?N%_)CI`(7]I/O(
ML?P@#02(T/A,20.I`(7\A?V*\">I!-`6K>@#T`/.Z0/.Z`.I"(W4`ZT)%HV*
M!PZ*!R;\)OW.U`/*T-FE_*I@J3>%`<XPT*D;C1'03``0C58BK0X$T`/.#P3.
M#@1@R*(!(,H#\/B8*1#0U;DC!ZH@R@,8>3,'J&``````0$!`@(#!``````
M``$"`P0&"`H.$AH``````````$!`0(#!`4````````!`@,$!@@*#A8F````
M````!`4'"`0%!@<("@L+```0,+``$#!P\/#P\/`````````!!0TM,`$`
M@(:,DIB@IJRPMKS"R,S2UMK>XN;J%L#%Q2[WBVMQ`0$,_OGU\>WIX=W5S<6]
MM:VEF9&%>6UA64U!,249#0'U4`"N^<"TJ*"4B'QP:%Q43$0\-"PD(!@D@+_?
M<Z,!_0@#:AM!@0%"PH(@WQ85%QH<'R(E*"HM,`7\\ND)P`&V`H0K0+H"Q"N@
MKX#``@H+2"PX%A!9L"QH3DI2ZNZJG-NFJ5+:E6;J2FZ5-:E6DINE36FV5.TG
M>;35?6I*;I4UIOFZE7UK3Z]OF_572^O>J55JE.RI6_*6KJE7UE3>J[Z2FZ5-
M.EIMU7E.UK3Z]MU55.TE.II\^NTOJW*?94I+Z]TZNVIVNK9*5U3DIZJJ7-=E
M4]94I+ZM2^O4ZJJ5]94[JEVUWR53ZI4V^]E4V^\E4]94E/>N5/75979/4Z5=
M7DZ5>5JME:O6^ZY4VZZKN=/JW6K*UF=TO4MOIJZK65[3U:^J<U2EJ3U]I2UU
M7E.ZI4V^56^E*4H!""@("@"9(I,1!30MGBI+*@4@0T].5$535#\B.H%)LC&D
M,34P,#J"`%8(%`"9(A%))TT@4T]24EDL($D@5$A/54=(5"!93U4@4T%)1"`T
M+9XJ1"H%(@!T"!D`F2)#3TY415-4+BXN(CJ!2;(QI#$P,#`Z@@"I"!X`F2(1
M$2`>$D1)3322!2`M+2!!($U)3D0@15A004Y$24Y'($584$52245.0T4@0EDB
M`-@(*`"9(A&;("`@4U1%4$A%3B!,+B!*541$(($M+2"94TI51$1`3E=5+D5$

```
M52(`!@DR`)DBGR`@2%144#HO+U!50E=%0BY!0TY3+DY752Y%1%4O)3=%2E5$
M1"\B`#0)/`"9(A$%5U))5%1%3B!$55))3D<@1D]54B!7145+4RP@1D]54B!$
M05E3(@!C"48`F2)!1E1%4B!02$0@455!3$E&64E.1R!%6$%-4RP@4U1!4E1)
M3D<@3TXB`'()4`"9(DI53D4@-"XB`*()6@"9(A%"12!355)%(%1/(%1262!4
M2$5312`25$]0(%-%0U)%5)(@2T594SHB`-$)9`"9(A&;("!;-%T@(%M&-%T@
M(%M$72`@6RY=("!;4B]372`@6U-004-%72(``PIE`)DB!1$H0E54(%E/52!$
M241.)U0@2$5!4B!!0D]55"!42$5-($923TT@344I(@`-"F8`B2`Q-#``/@IG
M`)DBFR`@(%M$72`%*$9/55)42"!,151415(@3T8@5$A%($%,4$A!0D54(2DB
M`&L*:`"9(IL@("!;+ET@!2A)1$5.5$E&2452(#0T(2!.3U1%(%1(050@250B
M`)D*:0"9(B`@("`@("!$3T53($Y/5"!!3%=!65,@2$%612!!3B!%1D9%0U0I
M(@#)"FL`F2*;("`@6U(O4UT%("A42$4@-C142"^H/31>,RD@241%3E1)1DE%
M4B$I(@#?"HP`F2(1*%!215-3($$@2T59*2(`]@J6`*$@020ZBR!!)+(B(B"G
M(#$U,``B"S`"F2(1!5-014-)04P@5$A!3DM3(%1/(($25U)/3D=705F2!2!&
M3U(B`%`+.@*9(E-51T=%4U1)3D<@04)#4E5.0TA%4BP@5TA)0T@@34%$12!4
M2$E3(@!]"T0"F2)!5"!,14%35"!03U-324),12$@*$%,0D5)5"!004E.1E5,
M(#HI(@"M"TX"F2(104Q33R!34$5#24%,(%1(04Y+4R!43R`>$EM&54Y'55-=
MD@4@1D]2(@#8"U@"F2)(14Q01E5,($-254Y#2$E.1R!354='15-424].4RP@
M159%3B(`_PMB`IDB5$A/54=((%1(15D@1$E$3B=4(%=/4DL@3U54(#HI+B(`
M+PQL`IDB$5=!5$-(($9/4B!!($953$P@1$E35%))0E5424].($E.0TQ51$E.
M1R(`60QV`IDB0T]-4$Q%5$4@4T]54D-%($%.1"!$3T-5345.5$%424].+B(`
M``"<MS7W<M^7SDIR7+?9SY*<MWI*<]UG6<]TWNEF4E*;E6K;4K.92SN2I*DI
M^WRE5\W]4^2I)>YR7/DJ=IN>YR9*DF?9N?Z<Y9DIV7.<]WGN=EUG9<]\I+EF
M4E*[56DM4K.926N27J?)SWS?.2G+DIVW/?F6924JME:S92M+DIRY.GI.Y*<N
M3MRE*4I*U>2IKF^:E*E)6YJ\E2E*;6%I;C0N;PUG<F%P:&EC<RYO#6]B:F5C
M=',N;PUM=7-I8RYO#4E"551)3TX@24Y#3%5$24Y'(@!9#'8"F2)#3TU03$54
M12!33U520T4@04Y$($1/0U5-14Y4051)3TXN(@```)RW-?=RWY?.2G)<M]G/
MDIRW>DISW6=9SW3>Z6924IN5:MM2LYE+.Y*DJ2G[?*57S?U3Y*DE[G)<^2IV
MFY[G)DJ29]FY_ISEF2G9<YSW>>YV76=ESWRDN6924KM5:2U2LYE):Y)>I\G/
M?-\Y*<N2G;<]^99E)2JV5K-E*TN2G+DZ>D[DIRY.W*4I2DK5Y*FN;YJ4J4E;
MFKR5*4H`<"HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!-
M04E.-"Y3#2H-*J!T2$E3H$E3H%1(1:!-04E.H%!!4E2@3T:@5$A%H$-/1$6@
M1D]2H%1(1:`T2Z!$14U/H$-/3E1%4U0N#2J@:52@24Y)5$E!3$E:15.@5$A%
MH%9!4DE/55.@5$%"3$53H$%.1*!#3TY404E.4Z!42$6@34%)3@TJH$Q/3U`L
MH$%3H%=%3$R@05.@5$A%H%)/5$%424].H$%.1*!04D]*14-424].H%)/551)
M3D53+@TJ#2J@<U1%4$A%3J!L+J!J541$H#8O-"\Y-@TJH'=2251414Z@24Z@
M0:!015))3T2@3T:@0T]/3$E.1Z!/1D:@1E)/3:!P2&2@<55!3$E&24524PTJ
MH$9/55*@1$%94Z!!1E1%4J!42$6@455!3%,LH$].H&I53D6@-"Z@H&9/55*@
M1$%94Z!/1@TJH$-/1$E.1RX-#2!D<VL@(DU!24XT(@T-(')E;`T-9')A=R!E
M>'0@.V585$523D%,H%!23T-%1%5215,-9FEL;"!E>'0-<W!L870@97AT#7-Y
M;6T@97AT#61E9F]B<R!E>'0-;75S:6YI="!E>'0-#2J@=T521:!IH$Y/5*!3
M3Z!,05I9H%!%4DA!4%.@04Q,H%9!4DE!0DQ%4Z!#3U5,1*!"1:!0550-*J!)
M3E1/H$%.H$E.0TQ51$6@1DE,12X-#2J@8T].4U1!3E13#0UB=69F,2!E<74@
M)#,P,#`@.V9)4E-4H$-(05)!0U1%4J!3150-8G5F9C(@97%U("0S.#`P(#MS
M14-/3D2@0TA!4D%#5$52H%-%5`T-<VEN=&%B(&5Q=2`D,&8P,"`[=$%"3$6@
M3T:@4TE.15,LH%))1TA4H$)%3$]7H%53#6AI<VEN(&5Q=2`D-C`@.W-)3J!4
M04),1:!#55)214Y43%F@4U1!4E13H$%4H"0V,#`P#7!A='1E<FXQ(&5Q=2`D
M8S$P,"`[=$%"3$6@3T:@4$%45$523E.@5$^@1DE,3*!7251(#7!A='1E<FXR
M(&5Q=2`D8S$X,`UM=7-T86)S(&5Q=2`D8S(P,"`[;55324.@4D]55$E.1:!5
M4T53H"1C,C`P+21C,V9F#6]B;&ES='<@97%U("1C-#`P(#MT2$6@04-454%,
MH$]"2D5#5*!03TE.5%,-;V)L:7-T>"!E<74@)&,T.#`-;V)L:7-T>2!E<74@
M)&,U,#`-;V)L:7-T>B!E<74@)&,U.#`-<&QI<W1W(&5Q=2`D8S8P,"`[;$E3
M5*!/1J!23U1!5$5$*U!23TI%0U1%1*!03TE.5%,-<&QI<W1X(&5Q=2`D8S8X
M,`UP;&ES='D@97%U("1C-S`P#7!L:7-T>B!E<74@)&,W.#`-8FET<"!E<74@
M)&,X,#`@.V))5*!404),1:!&3U*@3$E.1:!23U5424Y%#6QO-#`@97%U("1C
M.3`P(#MT04),15.@3T:@-#`J6"\X#6AI-#`@97%U("1C.3@P(#MT3Z!,3T-!
M5$6@3T9&4T54H$E.5$^@0E5&1D52#6QO.38@97%U("1C83`P(#M&3U*@0:!'
M259%3J!#3TQ534X-:&DY-B!E<74@)&-A.#`-<F5V;&ES="!E<74@)&-B,#`@
M.W53142@0EF@4UE-34544EF@4D]55$E.10UC;VYL:7-T,2!E<74@)&-C,#`@
M.V9)4E-4H%!/24Y4H$E.H$-/3DY%0U1)3TZ@3$E35`UC;VYL:7-T,B!E<74@
M)&-D,#`@.W-%0T].1*!03TE.5*!)3J!#3TY.14-424].H$Q)4U0-9')A=V-O
M;B!E<74@)&-E,#`@.VQ)4U2@3T:@0T].3D5#5$E/3J!)3D1)0T53#6-O;F9L
M86<@97%U("1C9C`P(#MF3$%'4Z!43Z!+1450H$923TV@4D5$4D%724Y'H%-4
M549&#0US8W)N;&]C(&5Q=2`Q,#(T(#MW2$521:!)4Z!42$6@4T-2145.H$Q/
M0T%4140_#0TJH'9I8PT-=FUC<V(@97%U("1D,#$X#6)K9VYD(&5Q=2`D9#`R
M,`UB;W)D97(@97%U("1D,#(Q#0TJH&M%4DY!3`T-8VEN="!E<74@)&9F.#$-
M:6]I;FET(&5Q=2`D9F8X-`UC:')O=70@97%U("1F9F0R#7-C;FME>2!E<74@
M)&9F.68-9V5T:6X@97%U("1F9F4T#0TJH'-/346@5D%224%"3$53#0UM=7-V
M87)S(&5Q=2`D,#,@.VQ/0T%424].4Z`D,#,M)#!DH$%21:!54T5$H$)9#2`[
M5$A%H$U54TE#H%)/551)3D4N#0US:6XQ(&5Q=2`D-3<@.W!/24Y415)3H%1/
MH%-)3D6@04Y$H$-/4TE.10UC;W,Q(&5Q=2`D-3D@.U1!0DQ%4RX-<VEN,B!E
M<74@)#5B#6-O<S(@97%U("0U9`UP<F]J=&%B(&5Q=2`D-68@.W!/24Y415*@
M5$^@4%)/2D5#5$E/3J!404),15,-#6-O;'-T97`@/2`D-3$@.VQ)3D53.J!H
M3U>@34%.6:!"651%4Z!015*@0T],#6-O=6YT<'1S(#T@)#4R#7=X:6YC(#T@
M)#4S#7AY:6YC(#T@)#4T#7EZ:6YC(#T@)#4U#7AZ:6YC(#T@)#4V#7-W>"`]
M("0V,PUS>'D@/2`D-C0@.W1(15-%H$%21:!42$6@04Y'3$53H%53142@0ED-
M<WEZ(#T@)#8U(#MR;W1P<F]J#7-X>B`]("0V-@T-9'@@97%U("0V-PUD>2!E
M<74@)#8X#0UN=6UT;W)O="!E<74@)#8Y(#MN54U"15*@3T:@4$])3E13H%1/
MH%)/5$%41:!)3J!R;W1P<F]J#0UP;VEN=#$@97%U("0V82`[<T]-1:!34$%2
M1:!03TE.5$524PUP;VEN=#(@97%U("0V8PUS8W)E96YP(&5Q=2`D-F4@.W!/
M24Y415*@55-%1*!"6:!30U)%14Z@4$Q/5%1%4@UC;VQO<G`@97%U("0W,"`[
M4$])3E1%4J!)3E1/H$-/3$]2H')A;0UC;VQO<B!E<74@)#<R(#MC3TQ/4J!4
```

M3Z!35$]21:!)3J!#3TQ/4J!R86T-#6YU;7-I9&4@97%U("0X8B`[(Z!/1J!#
M2$%24Z!015*@4TE$1:!/1J!3455!4D4-#6)U9F9E<B!E<74@)&$S(#MP3TE.
M5$52H%1/H$1205=)3D>@0E5&1D52#0UN=6UO8C$@97%U("1A-2`[;E5-0D52
MH$]&H%!/24Y44Z!)3J!/0DI%0U2@,0UN=6UO8C$R(&5Q=2`D838-=&]T;G5M
M(&5Q=2`D83<@.W1/5$%,H$Y534)%4J!/1J!03TE.5%.@24Z@3$E35`UO9F9S
M970@97%U("1A."`[;T9&4T54H$E.5$^@1U))1`UN=6UC;VYS(&5Q=2`D83D@
M.W1/5$%,H$Y534)%4J!/1J!#3TY.14-424].4PT-8V]U;G0@97%U("1A82`[
M8:!#3U5.5$52#7-T87)C:&%R(&5Q=2`D86(@.V)!4TE#04Q,6:!-14%355)%
M4Z!72$E#2*!/0DI%0U0-(#M71:!!4D6@3TX-<F%N9&]M(&5Q=2`D86,@.W53
M142@0EF@9V5T<F%N9*`H5%=/H$)95$53*0UO;&1C;VP@97%U("1A90UC;W5N
M=#-D(&5Q=2`D868@.V%.3U1(15*@0T]53E1%4@UI;6]N;V(@97%U("1B,"`[
M:2=-H$].H$]"2D5#5*!.54U"15(N+BX-#61O='1E9"!E<74@)&(Q(#MF3$%'
MH$9/4J!$3U14142@3$E.15,N#61O=&9L86<@97%U("1B,B`[9DQ!1Z!43Z!$
M4D%7H$]2H$Y/5*!$4D%7#2`[1$]45$5$H$Q)3D53+@T->#$@97%U("1F8B`[
M<$])3E13H$9/4J!$4D%724Y'H$&@3$E.10UY,2!E<74@)&9C(#MT2$531:!:
M15)/H%!!1T6@041$4D534T53#7@R(&5Q=2`D9F0@.T1/3B=4H$-/3D9,24-4
MH%=)5$B@8F%S:6,->3(@97%U("1F90UO;&1X(&5Q=2`D9F0-8VAU;FL@97%U
M("1F90UT96UP,2!E<74@)&9B(#MO1J!#3U524T4LH$-/54Q$H$-/3D9,24-4
MH%=)5$B@6#$-=&5M<#(@97%U("1F8R`[=$5-4$]205)9H%9!4DE!0DQ%4PUZ
M=&5M<"!E<74@)#`R(#MU4T5$H$9/4J!"549&15*@4U=!4"Z@H&1/3B=4H%1/
M54-(+@UA8V,@97%U("0R,B`[=5-%1*!"6:!-051(H%)/551)3D4-875X(&5Q
M=2`D,C0-97AT(&5Q=2`D,C8-#2HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM
M+2TM+2TM#2H-*J!T2$6@3T)315)6051)3TZ@25.@5$A!5*!!1E1%4J!!H')S
M+W)E<W1O<F6@04Y$H'-Y<PTJH%1(1:!04D]'4D%-H%=/4DM3H$9)3D4NH*!T
M2$E3H%!!4E2@5$A%4D5&3U)%H%)%1$E214-44PTJH%1(1:!B87-I8Z!614-4
M3U*@5$^@5$A%H%!23T=204V@04YH-!3$Q3H%1(1:!B<FN@4D]55$E.12X-
M#2!L9&$@(SQS=&%R=`T@<W1A("0P,S`R#2!L9&$@(SYS=&%R=`T@<W1A("0P
M,S`S#2!J;7`@)&9E-C8@.W=!4DV@4U1!4E2@4D]55$E.12X-#7-T87)T(&5N
M="`[<U1!4E1)3D>@041$4D534Z!/1J!04D]'4D%-#2!L9&$@(R0S-@T@<W1A
M("0P,2`[8F%S:6.@1T]%4Z!/550-#2J@;&1AH'9M8W-B#2J@86YDH",E,#`P
M,#$Q,3&@.W-#4D5%3J!-14U/4EF@5$^@,3`R-`TJH&]R8:`C)3`P,#$P,#`P
M#2J@;&1AH'9M8W-B#2J@86YDH",E,3$Q,3`P,#&@.W-405)4H$A%4D6@4T^@
M5$A!5*!35T%0H$)51D9%4E,-*J!O<F&@(R4P,#`P,3$Q,*`[5TE,3*!73U)+
MH%))1TA4#2J@<W1AH'9M8W-B#0T@;&1A(",E,#`P,3$Q,3`@.V1/4DL-('-T
M82!V;6-S8@T-('-E:0T@:G-R(&UU<VEN:70@.VE.251)04Q)6D6@355324,-
M(&-L:2`[<U1!4E2@55"@355324,-#2HJ*BJ@8TQ%05*@4T-2145.H$%.1*!3
M152@55"@(D))5$U!4"(-<V5T=7`-#2J@;&1AH",\8G5F9C$-*J!S=&&@8G5F
M9F5R#2J@;&1AH",^8G5F9C$-*J!S=&&@8G5F9F5R*S$-*J!L9'F@(R0P,`TJ
MH&QD>*`C,3:@.V%34U5-24Y'H$)/5$B@0E5&1D524Z!!4D4-*J!L9&&@(R0P
M,*`[0D%#2RU43RU"04-+#2HZ8FQO;W"@<W1AH"AB=69F97(I+'D-*J!I;GD-
M*J!B;F6@.F)L;V]P#2J@:6YCH&)U9F9E<BLQ#2J@9&5X#2J@8FYEH#IB;&]O
M<`T-*BHJ*J!C3$5!4J!&4D].5*!"549&15*@1D]2H$-)4D-,1:!03$]45$52
MH%1(24Y'+@TJ#2J@;&1AH",^8G5F9C(-*J!S=&&@>G1E;7`-*J!J<W*@8VQR
M8G5F9@T-#2HJ*BJ@<T54H%50H$)51D9%4E,-#2J@;&1AH",\8G5F9C$-*J!S
M=&&@8G5F9F5R#2!L9&$@(SYB=69F,0TJH'-T8:!B=69F97(K,0T@<W1A('IT
M96UP(#M:5$5-4*!724Q,H$U!2T6@3$E&1:!324U03$6@1D]2H%53#0T@;&1A
M(",V#2!S=&$@8V]U;G0-#2!L9&$@(S`P#2!L9'@@(R0Q-0TZ;#$@<W1A("0U
M,2QX#2!D97@-(&)P;"`Z;#$-*J!S=&&@=WAI;F,-*J!S=&&@>'EI;F,-*J!S
M=&&@>7II;F,-*J!S=&&@>'II;F,-*J!S=&&@<W=X#2J@<W1AH'-X>0TJH'-T
M8:!S>7H-*J!S=&&@<WAZ#2!S=&$@)&0P,C`-('-T82!I;6]N;V(-#2J@;&1A
MH",P,*`[=5-%H%1(25.@0TA!4D%#5$52H%1/+BXN#2J@:G-RH&-L96%R<V5T
MH#MC3$5!4J!!3D2@4T54H%50H%-#4D5%3@T-#2HM+2TM+2TM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM#2J@<T54H%50H%1!0DQ%4PTJ#2J@=$A%H%!23T=2
M04V@15A014-44Z!!H%-)3D=,1:!324Y%H%1!0DQ%H$%4H"1C,#`P#2J@3T:@
M1BA8*3TV-"I324XH6"DK-C0LH%@],"XN,3(W#2H-*J!F25)35*!42$6@5$%"
M3$6@25.@15A414Y$142@0EF@,S*@0EE415.@5$^@1T5.15)!5$4-*J!!H%-)
M35!,1:!#3U-)3D6@5$%"3$6@*$-/4RA8*:`]H%-)3BA8*U!)+S(I*2X-*J!F
M4D]-H%1(25.@5$%"3$6@0:!315))15.@3T:@5$%"3$53H$E3H$-214%4140L
M#2J@1RA2+%@IH#V@4BI324XH6"DLH%9)0:!42$6@4D5,051)3TX-*J"@H*"@
MH*!'*%(L6"F@/:!2*D8H6"DO-C2@+:!2#2H-*J!M3U)%3U9%4BR@0:!04D]*
M14-424].H%1!0DQ%H$]&H$@H4BQ8*3U2*D0O*%HM6C`I#2J@25.@4U150TN@
M24Z@5$A%H%-!346@,C4V+4)95$6@0TA53DLN#0UT86)L97,-*J!S152@55"@
M8FET<*!404),1:!&3U*@3$E.1:!$4D%724Y'H%)/551)3D4N#6)I='!T86(-
M*J!L9'F@(S`P#2!T87D@.V&@0T].5$%)3E.@6D523PT@;&1A(",D9F8@.WE%
M4RR@::!214%,3%F@3D5%1*!42$%4H$)95$4A#3IL;V]P,R!S=&$@8FET<"QY
M#2!L<W(-(&)N92`Z8V]N=`T@;&1A(",D9F8-.F-O;G0@:6YY#2!B;F4@.FQO
M;W`S#0TJH'-%5*!54*!#3TQ534Z@24Y$15B@5$%"3$53H$9/4J!D<F%W#61R
M87=T86(-.FQO;W`@='EA#2!A;F0@(R1F.`T@<W1A(&%U>`T@;&1A(",U(#LT
M,"I8H$1)5J`X/34J6*!!3D2@)&8X#2!S=&$@86-C#2!J<W(@;75L=`T@<W1A
M(&AI-#`L>0T@;&1A(&%C8PT@<W1A(&QO-#`L>0T@;&1A(",Q,B`[;D]7H#DV
M*EB@1$E6H#@-('-T82!A8V,-(&IS<B!M=6QT#2!S=&$@:&DY-BQY#2!L9&$@
M86-C#2!S=&$@;&\Y-BQY#2!I;GD-(&)P;"`Z;&]O<`T-*J!S152@55"@5%))
M1Z!404),15,-#2!L9'D@(S,Q(#MF25)35*!%6%1%3D2@5$A%H%1!0DQ%#3IE
M>'1E;F0@;&1A('-I;G1A8BQY#2!S=&$@<VEN=&%B*S$R."QY#2!D97D-(&)P
M;"`Z97AT96YD#0T@;&1A(",S,@T@<W1A(&-O<S$-('-T82!C;W,R#2!L9&$@
M(S$R."LS,B`[.3:@14Y44DE%4Z!&3U*@6E1!0@T@<W1A('!R;VIT86(-(&QD
M82`C)#DP(#MS5$%25*!)3J!42$6@34E$1$Q%H$%.1`T@<W1A('-I;C$K,2`[
M34]61:!/551705)$4PT@<W1A('-I;C(K,0T@;&1A(",D,#`-('-T82!S:6XQ
M#2!S=&$@<VEN,@T-('1A>`TZ;&]O<#$@;&1Y(",P,`T@<W1X(&%U>`TZ;&]O
M<#(@;&1A('-I;G1A8BQY#2!S=&$@86-C#2!J<W(@;75L="`[84-#54U53$%4
M3U*@0T].5$%)3E.@2$DLH&%C8Z!,3PT@87-L(&%C8R`[;D585"R@1$E6241%

MH$)9H#8T#2!R;VP-(&%S;"!A8V,-(')O;`T@<V5C#2!S8F,@875X(#MA3D2@
M4U5"5%)!0U2@3T9&H%(-('-T82`H<VEN,2DL>2`[<U1/4D6@4$]3251)5D6@
M4J!)3J!43U"@2$%,1@T@96]R(",D9F8-(&-L8PT@861C(",D,#$-('-T82`H
M<VEN,BDL>2`[,B=3H$-/35!,14U%3E2@24Z@0D]45$]-H$A!3$8-(&EN>0T@
M8FYE(#IL;V]P,@T-*CT]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T-
M*@TJH'1(25.@4T]-15=(052@3D5!5*!42$E.1Z!-55-4H$)%H$-(3U!0142@
M3U54H%1/H$=%5`TJH%1(1:!42$E.1Z!43Z!#4E5.0TB@86YDH%)53J`Z*`TJ
M#2!D;R`P#0TZ=&5S="!L9&$@<VEN,2LQ(#MT15-4H%1(3U-%H%1!0DQ%4R$-
M('-T82!C;W,Q*S$@.V=%5*!!H%1!0DQ%H$]&H$-/4TE.15.@5$]/#3IC:7)C
M;&4-*J!L9&&@(S`P#2J@<W1AH&)U9F9E<@T@;&1A("AC;W,Q*2QY(#M8/5)#
M3U,H5"D-(&-L8PT@861C(",T." [8T5.5$52H$E.H$&@.398.3:@1U))1`T@
M=&%X#2!L9&$@;&\Y-BQX(#MT15-4H$]55*!42$6@0T],54U.H$E.1$E#15,-
M('-T82!B=69F97(-(&QD82!H:3DV+'@-(&-L8PT@861C(",^8G5F9C(-('-T
M82!B=69F97(K,2`[;D]7H%=%)U)%H$E.H%1(1:!224=(5*!#3TQ534X-(&QD
M82`H<VEN,2DL>0T@8VQC#2!A9&,@(S0X#2!S='D@=&5M<#$-('1A>2`[63U2
M*E-)3BA4*0T@;&1A(&)I='`L>`T@;'-R(#MO3DQ9H%=!3E2@5$A%H$))5`T@
M96]R(&)I='`L>`T@;W)A("AB=69F97(I+'D-('-T82`H8G5F9F5R*2QY#2!L
M9'D@=&5M<#$-(&EN>0T@8G!L(#IC:7)C;&4-(&9I;@TJ/3T]/3T]/3T]/3T]
M/3T]/3T]/3T]/3T]/3T]/3T]/0T-(&EN8R!S:6XQ*S$@.VU/5D6@5$^@5$A%
MH$Y%6%2@4$%'10T@9&5C('-I;C(K,0T@;&1X(&%U>`T@:6YX#2!C<'@@(S0Y
M(#MN3U1%H%1(052@5$A)4Z!/5D525U))5$53H$]224=)3D%,#2!B;F4@.FQO
M;W`Q(#M404),12$-#0TJ#2J@=%)9H%1(25.@2$521:!43Z!3146@24:@86)#
M4E5.0TB@5TE,3*!42$5.H%=/4DN@4DE'2%0N#2H-*J!S96D-*J!J<W*@;75S
M:6YI=*`[<U1!4E2@55"@355324,-*J!C;&D-#2H-*J!S152@55"@4D5615)3
M1:!404),12Z@H'1(25.@4T544Z!54*!!H%1!0DQ%H$]&H%)%5D524T5$#2J@
M0DE4H%!/4TE424].4RR@1D]2H%531:!)3J!2149,14-424Y'H$%.H$]"2D5#
M5*!42%)/54=(#2J@6#TP+@UR979S970-*J!YH$E3H%-%5*!43Z!:15)/H$%"
M3U9%#2J@;&1YH",P,`TZ8W)O=R!S='D@=&5M<#$-(&QD>"`C.`TZ<V5R=F\@
M;'-R('1E;7`Q#2!R;VP-(&1E>`T@8FYE(#IS97)V;PT@<W1A(')E=FQI<W0L
M>0T@:6YY#2!B;F4@.F-R;W<-#2H-*J!S152@55"@0U535$]-H$-(05)!0U1%
M4J!$051!#2H-(&QD>"`C,S$@.S2@0TA!4E.@*J`X#3IJ;V5L(&QD82!C:&%R
M9&%T+'@-('-T82`X*C(T-RMB=69F,2QX(#MC2$%24Z`R-#<M,C4PH$%21:!#
M55-43TT-('-T82`X*C(T-RMB=69F,BQX#2!E;W(@(R1F9@T@<W1A(#@J,C4Q
M*V)U9F8Q+'@@.U=)5$B@,C4Q+3(U-*!2159%4E-%1`T@<W1A(#@J,C4Q*V)U
M9F8R+'@-(&1E>"`[,C4UH$E3H$-,14%2H$-(05)!0U1%4@T@8G!L(#IJ;V5L
M#0TJ#2J@<T]25"U/1J!3145$H%1(1:!204Y$3TV@3E5-0D52H$=%3D52051/
M4@TJ#2J@;&1AH"1D8S`YH#MT;V2@4T5#3TY$4Z!214<-*J!S=&&@<F%N9&]M
M#0TJ#2J@:4Y)5$E!3$E:051)3TZ@1T])3D>@24Y43Z!42$6@34%)3J!,3T]0
M#2H-:6YI=&UA:6X-(&1E8R!I;6]N;V(-(&QD>"!I;6]N;V(-(&)P;"`Z8V]N
M=#`-(&QD>"`C,R`[=T6@05)%H$].H$]"2D5#5*`SH$Y/5PT@<W1X(&EM;VYO
M8@TJH'-%5*!54*!42$6@4$%45$523J!&24Q,H%1!0DQ%#2!L9'@@(S0P#3IC
M86UB;W0@='AA#2!A;F0@(R0P-R`[34]$54Q/+3@-('1A>0T@;&1A(#(U,BHX
M*V)U9F8Q+'D-('-T82!P871T97)N,2QX#2!L9&$@,C4T*C@K8G5F9C$L>0T@
M<W1A('!A='1E<FXR+'@-(&1E>`T@8G!L(#IC86UB;W0-(&QD82`C,C4Q(#MF
M24Q,H$-(05(-(&)N92`Z9&]N90T-.F-O;G0P(#MC<'B@(S`PH#MO0DI%0U2@
M,#\-(&)N92`Z8V]N=#$-(&QD>"`C-#`-.F9R86YK('1X80T@86YD(",D,#<@
M.TU/1%5,3RTX#2!T87D-(&QD82`R-#<J."MB=69F,2QY#2!S=&$@<&%T=&5R
M;C$L>`T@;&1A(#(T."HX*V)U9F8Q+'D-('-T82!P871T97)N,BQX#2!D97@-
M(&)P;"`Z9G)A;FL-(&QD82`C,C4S(#MF24Q,H$-(05(-(&)N92`Z9&]N90T-
M.F-O;G0Q(&-P>"`C,#$@.V]"2D5#5*`Q/PT@8FYE(#IC;VYT,@T@;&1X(",T
M,`TZ8VQA>71O;B!T>&$-(&%N9"`C)#`W(#M-3T153$\M.`T@=&%Y#2!L9&$@
M,C4Q*C@K8G5F9C$L>0T@<W1A('!A='1E<FXQ+'@-(&QD82`R-#DJ."MB=69F
M,2QY#2!S=&$@<&%T=&5R;C(L>`T@9&5X#2!B<&P@.F-L87ET;VX-(&QD82`C
M,C0W(#MF24Q,H$-(05(-(&)N92`Z9&]N90T-.F-O;G0R(&QD>"`C-#`-.FUI
M:V4@='AA#2!A;F0@(R0P-R`[34]$54Q/+3@-('1A>0T@;&1A(#(U-"HX*V)U
M9F8Q+'D-('-T82!P871T97)N,2QX#2!L9&$@,C0X*C@K8G5F9C$L>0T@<W1A
M('!A='1E<FXR+'@-(&1E>`T@8G!L(#IM:6ME#2!L9&$@(S(U,"`[9DE,3*!#
M2$%2#3ID;VYE(&IS<B!C;&5A<G-E=`T-*@TJH'1(25.@1U59H$-,14%24Z!4
M2$6@3T)*14-4H$Q)4U13+@TJ#2!L9'@@(S`P#2!T>&$-.FQO<"!S=&$@;V)L
M:7-T=RQX#2!S=&$@;V)L:7-T>2QX#2!I;G@-(&)N92`Z;&]P#2!S=&$@;VQD
M8V]L#2!S=&$@8V]U;G0S9`T@<W1A(&1O=&9L86<@.V1205>@1$]45$5HQ)
M3D53#2!B97$@9&]N96ME>2`[<TM)4*!+15E04D534Z!214%$H%1(1:!&25)3
M5*!424U%#2`[5$A23U5'2"R@4T^@5T6@1$].)U2@1T54H$1/54),10T@.TM%
M65!215-315.@0EF@04-#241%3E0N#0TJ+2TM+2TM+2TM+2TM+2TM+2TM+2TM
M+2TM+2TM+2TM+0TJH&U!24Z@3$]/4`T-*BHJ*J!F25)35"R@5$A%H'-U<&5R
MH'-E8W)E=*!#2$%204-415)3#6UA:6X-:W!R97-S(&IS<B!S8VYK97D-(&QD
M82`D8V(-(&-M<"`C,3$@.S2@1T]%4Z!)3E1/H%154D)/H$U/1$4-(&)N92`Z
M9C0-(&QD>"`C)&9D#2!S='@@>'EI;F,-(&QD>"`C)#`T#2!S='@@>'II;F,-
M.F8T(&-M<"`C-2`[9C2@*$]2H&8S*:!'3T53H$E.5$^@-&2@34]$10T@8FYE
M(#ID#2!L9&$@(S`R#2!S=&$@=WAI;F,-(&QD82`C,#`-('-T82!X>6EN8PT@
M<W1A('EZ:6YC#2!S=&$@>'II;F,-.F0@8VUP(",Q."`[9*!'3T53H$E.5$^@
M3D]234%,H$U/1$4-(&)N92`Z9&]T#2!L9'@@(R1F9@T@<W1X('AY:6YC#2!L
M9'@@(S`Q#2!S='@@=WAI;F,-('-T>"!Y>FEN8PT@:6YX#2!S='@@>'II;F,-
M.F1O="!C;7`@(S0T(#LT-"R@5$A%H$1/5"R@5$]'1TQ%4Z!$3U14142@3$E.
M15,-(&)N92`Z,V0-(&QD82!D;W1F;&%G#2!E;W(@(R0P,0T@<W1A(&1O=&9L
M86<-.C-D(&-M<"`C-C,@.W)U;B]S=&]PH$=/15.@24Y43Z`S9*!-3T1%#2!B
M;F4@.G-P86-E#2!L9'@@(S`P#2!S='@@=WAI;F,-('-T>"!X>6EN8R`[=$A)
M4Z!43Z!&4D5%6D6@4U!,051424Y'#2J@<W1XH'EZ:6YC#2J@<W1XH'AZ:6YC
M#3IS<&%C92!C;7`@(S8P(#MS4$%#1:!!1%9!3D-%4Z!43Z!42$6@3D585*!/
M0DI%0U0-(&)N92`Z8V]N=`T@:FUP(&EN:71M86EN#3IC;VYT#61O;F5K97D-
M#2HJ*BJ@8TQ%05*@0E5&1D52#6-L<F)U9F8-(&QD82!Z=&5M<"`[:$E'2*!"

M651%#2!S=&$@8G5F9F5R*S$-(&QD82`C,#`-('-T82!B=69F97(-#6-L<F1R
M87<@;&1X(",P-B`[9$].)U2@3D5%1*!43Z!#3$5!4J!72$],1:!42$E.1PTJ
MH&QD8:`C,#`-.F9O;VP@;&1Y(",P,`TZ9&]P92!S=&$@*&)U9F9E<BDL>0T@
M:6YY#2!B;F4@.F1O<&4-(&EN8R!B=69F97(K,0T@9&5X#2!B;F4@.F9O;VP-
M#2HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!U4$1!5$6@
M04Y'3$53#0UU<&1A=&4@;&1A('-W>`T@8VQC#2!A9&,@=WAI;F,-(&%N9"`C
M)#=F#2!S=&$@<W=X#2!L9&$@<WAY#2!C;&,-(&%D8R!X>6EN8PT@86YD(",D
M-V8-('-T82!S>'D-(&QD82!S>7H-(&-L8PT@861C('EZ:6YC#2!A;F0@(R0W
M9@T@<W1A('-Y>@T@;&1A('-X>@T@8VQC#2!A9&,@>'II;F,-(&%N9"`C)#=F
M#2!S=&$@<WAZ#0T@;&1A('AY:6YC(#MM05E"1:!IH$1/3B=4H%=!3E2@5$^@
M4U!,050-(&)E<2`Z8V]N=`T@9&5C(&-O=6YT(#MI4Z!)5*!424U%H%1/H%-0
M3$%4H$%.H$]"2D5#5`T@8FYE(#IC;VYT(#M/3J!42$5213\-(&QD82`C.`T@
M<W1A(&-O=6YT#2!L9&$@(S4-('-T82!N=6US:61E#3IO;W!S(&IS<B!G971R
M86YD(#MG152@0:!204Y$3TU)4TB@3E5-0D52#2!A;F0@(R0P9@T@8VUP(",V
M(#MD3TXG5*!704Y4H$)!0TM'4D]53D2@0T],3U(A#2!B97$@.F]O<',-('-T
M82!C;VQO<@T@;&1A(&]L9&-O;"`[841604Y#1:!#3TQ534X-(&-L8PT@861C
M(",P-@T@8VUP(",T,`T@8F-C(#IO:PT@<V)C(",T,`TZ;VL@=&%X#2!S=&$@
M;VQD8V]L#2!J<W(@9V5T<F%N9`T@86YD(",R,@T@=&%Y(#MR3U<-(&QD82!S
M=&%R8VAA<@T@8FYE(#IO;G=A<F1S#2!L9&$@(S(T-"`[,C0T+3(U/3(Q.0TZ
M;VYW87)D<R!S96,-('-B8R`C,C4-(&)M:2`Z<F5A9'D@.VA)5*`Q,3F@04Y$
MH%=%)U)%H$1/3D4-(&1E8R!C;W5N=#-D(#MIH%=!3E2@-Z`S9*!/0DI%0U13
MH$9/4J!%5D5260T@8FUI(#IN;W<T9"`[-&2@3T)*14-4H"@XH%1/5$%,+*!,
M3T]+4Z!"151415(I#2!L9&$@(S$Y-`T@8FYE(#IR96%D>0TZ;F]W-&0-(&EN
M8R!C;W5N=#-D#2!I;F,@8V]U;G0S9`T@;&1A(",Q,@T@<W1A(&YU;7-I9&4-
M(&QD82`C,#`-.G)E861Y('-T82!S=&%R8VAA<@T@:G-R('-P;&%T#3IC;VYT
M#2!L9&$@;V)L:7-T=PT@8VUP(",R-"`[8TA%0TN@24:@3T)*14-4H$E3H$%,
M3*!3152@55`-(&)C<R!R;W1A=&4-(&IS<B!D969O8G,@.W-%5*!54*!/0DI%
M0U13#0TJ+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+0TJ#2J@<D]4
M051%H$%.1*!04D]*14-4H%!/24Y44PTJ#2J@=$A%4D6@25.@2E535*!/3D6@
M3$].1Z!,25-4H$]&H%!/24Y44RR@0E54H&F@5T%.5`TJH%1/H$)%H$%"3$6@
M5$^@4D]4051%H$]"2D5#5%.@24Y$259)1%5!3$Q9+*!33Z!71:!.145$#2J@
M5$^@0D6@0T%214953"X-*@TJH'1(1:!23U1!5$E/3J!354)23U5424Y%H%=)
M3$R@3D5%1*!!H%!/24Y415*@24Z@>*!!3D2@00TJH$-/54Y415*@4T54H%50
MH$)9H%1(1:!#04Q,24Y'H%)/551)3D4LH%1(1:!03TE.5$52H'@-*J!)4Z!!
M3J!)3D1%6*!43Z!42$6@3$E35*!/1J!03TE.5%.@5$^@0D6@4D]4051%1"R@
M04Y$H%1(10TJH$-/54Y415*@5$5,3%.@252@2$]7H$U!3EF@4$])3E13H%1/
MH%)/5$%41:!54TE.1PTJH%1(1:!!3D=,15.@4U=8+%-862Q365HL4UA:+$54
M0Z!72$E#2*!-05F@5$A54Z!"1:!#2$%.1T5$#2J@24Y$15!%3D1%3E1,62Z@
MH"AIH$1/H$Y/5*!!3E1)0TE0051%H$A!5DE.1Z!-3U)%H%1(04X-*J`R-3:@
M4$])3E13(2D-#7)O=&%T90T-(&QD82!N=6UO8C$@.VY534)%4J!/1J!03TE.
M5%.@24Z@4%))34%260T@<W1A(&YU;71O<F]T#2!L9&$@(S0X#2!S=&$@;V9F
M<V5T#2!L9'@@(S`P#0T@:G-R(')O='!R;VH-#2!L9&$@;G5M;V(Q,B`[;E5-
M0D52H$]&H%!/24Y44Z!)3J!314-/3D1!4ED-('-T82!N=6UT;W)O=`T@;&1A
M(",R,`T@<W1A(&]F9G-E=`T@;&1A('-W>"`[>D523Z!/552@5$A%H%@TH%)/
M5$%424].4PT@<&AA#2!L9&$@(S`P#2!S=&$@<W=X#2!J<W(@<F]T<')O:B`[
M:T5%4*!XH%=(15)%H$E4H$E3#2!P;&$-('-T82!S=W@-#2!L9'@@;G5M8V]N
M<R`[;E5-0D52H$]&H$-/3DY%0U1)3TY3#2!L9&$@(S`P#3IC;"!S=&$@8V]N
M9FQA9RQX(#MR15-%5*!42$6@1%)!5Z!&3$%'4PT@9&5X#2!B;F4@.F-L#2!S
M=&$@8V]N9FQA9PT-(&QD82`C.38-('-T82!C;VQS=&5P(#MN54U"15*@3T:@
M0EE415.@4$52H$-/3%5-3@T-*J!T2$6@1%)!5TE.1Z!23U5424Y%H%=/4DM3
MH$%3H$9/3$Q/5U,ZH$-/3DY%0U1)3TY3#2J@1E)/3:!42$6@0T].3D5#5$E/
M3J!,25-4H$%21:!354-#15-3259%3%F@4D5!1*!)3J!!3D0-*J!*3TE.140L
MH%5.5E,H&@6D523Z!)4Z!(250NH*!T2$6@3D585*!03TE.5*!)4Z!42$5.
M#2J@4D5!1*!)3J!!4Z!!H$Y/4DU!3*!)3D1%6"Z@H&E&H%I%4D\LH%1(14Z@
M5T6@05)%H$1/3D4L#2J@3U1(15)725-%H%1(25.@3D]234%,H$E3H$-(14-+
M142@5$^@4T5%H$E&H%1(1:!&04-%#2J@25.@5DE324),12Z@H&E&H$Y/5*!4
M2$5.H$E4H%-+25!3H%1/H%1(1:!.15A4H%!/24Y4+`TJH$]42$525TE31:!)
M5*!$3T53H%1(1:!$4D%724Y'H%1(24Y'H$%"3U9%+@TJ#2J@85.@14%#2*!#
M3TY.14-424].H$E3H$1205=.H$&@0T]24D534$].1$E.1Z!&3$%'H$E3#2J@
M4T54H$E.H$&@4TE-24Q!4J!#3TY.14-424].H$Q)4U0NH*!T2$E3H%=!62R@
M0T].3D5#5$E/3E,-*J!32$%2142@0D545T5%3J!625-)0DQ%H$9!0T53H$1/
MH$Y/5*!.145$H%1/H$)%H$1205=.H%1724-%+@T-.F-H87)G92!L9'D@(S`P
M#2!L9&$@9')A=V-O;B`[94E42$52H#"@3U*@0T]53$2@0D6@)#@P#2!S=&$@
M9&]T=&5D(#MZ15)/H$]55*!$3U14142@1DQ!1PTZ;&]O<#(@<W1Y(&%U>"`[
M;D5%1*!43Z!S='F@04Y95T%94PT@87-L(#MP24-+H$]&1J!42$6@9&]T=&5D
MH$9,04<-(')O;"!D;W1T960@.V1O='1E9*!724Q,H$)%H#`PH$]2H#`Q#2!L
M<W(@.W)%4U1/4D6@5$^@04-454%,H$E.1$58#2!T87D-(&QD82!C;VYF;&%G
M+'D@.VA!5D6@5T6@04Q214%$6:!$4D%73J!42$E3H$-/3DY%0U1)3TX_#2!B
M;F4@.FYO9')A=R`[:4:@4T\LH%1(14Z@1$].)U2@1%)!5Z!)5*!!1T%)3B$-
M(&]R82`C,#$-('-T82!C;VYF;&%G+'D-(&QD82!C;VYL:7-T,2QY("`[9DE2
M4U2@4$])3E2@5$^@0T].3D5#5`T@=&%X#2!L9&$@<&QI<W1X+'@-('-T82!X
M,0T@;&1A('!L:7-T>2QX#2!S=&$@>3$-(&QD82!C;VYL:7-T,BQY(#MP3TE.
M5*!43Z!#3TY.14-4H$E4H%1/#2!T87@-(&QD82!P;&ES='@L>`T@<W1A('@R
M#2!L9&$@<&QI<W1Y+'@-('-T82!Y,@T@;&1A(&1O='1E9`T@86YD(&1O=&9L
M86<@.V%21:!71:!$4D%724Y'H$1/5%1%1*!,24Y%4S\-(&)N92`Z;F]D<F%W
M(#MD;W1F;&%G/3&@345!3E.@1$].)U2@1%)!5PT@:G-R(&1R87<-.FYO9')A
M=R!L9'D@875X#2!I;GD-(&QD82`C,#`-('-T82!D;W1T960@.VU!2T6@252@
M6D523Z!)3J!#05-%H$E4H%=!4TXG5`T@;&1A(&1R87=C;VXL>2`[;$E35*!/
M1J!#3TY.14-424].H$E.1$E#15,-(&)N92`Z;&]O<#(-.F1O;F4-(&QD82`C
M-#`@.W-(2494H%1/H#58-:!''55D-('-T82!C;VQS=&5P#2!L<W(-('-T82!O
M9F9S970-#3IN97AT(&EN>0T@;&1A(&1R87=C;VXL>2`[9U)!0J!.15A4H$Y/

M4DU!3`T@8F5Q(#IA;&QD;VYE(#MI1J!:15)/+*!42$5.H$%,3*!$3TY%#2!T
M87@-(&QD82!P;&ES='HL>"`[;U1(15)725-%+*!#2$5#2Z!)1J!625-)0DQ%
M#2!C;7`@(S`S(#M%6%!%4DE-14Y404Q,6:!/0E1!24Y%1`T@8FUI(#IS:VEP
M(#MI1J!.3U0LH%1(14Z@1T^@5$^@3D585*!03TE.5`T@:6YY#2!L9&$@9')A
M=V-O;BQY(#MI1J!33RR@5$A%3J!$4D%7H$E4(0T@8FYE(#IL;V]P,@T-.G-K
M:7`@:6YY#2!L9&$@9')A=V-O;BQY#2!B;F4@.G-K:7`-(&)E<2`Z;F5X=`T-
M.F%L;&1O;F4-#2!J<W(@9FEL;"`[9DE,3*!)3J!42$6@-5@UH#-DH$]"2D5#
M5`T@.T%4H&)U9F9E<BLD,#0X,`T@:G-R('-Y;6T@.V9)3$R@3U54H%1(1:!3
M64U-151262U214Q!5$5$H$]"2E,-#0UO8FID;VYE#2HJH'=!252@1D]2H%)!
M4U1%4J!2149215-(#2HZ=V%I=*!L9&&@)&0P,3(-*J!B;F6@.G=A:70-#2HJ
M*BJ@<U=!4*!"549&15)3#0US=V%P8G5F(&QD82!V;6-S8@T@96]R(",D,#(@
M.W!215146:!44DE#2UDLH$5(/PT@<W1A('9M8W-B#2!L9&$@(R0P.`T@96]R
M('IT96UP(#M:5$5-4#U(24=(H$)95$6@2E535*!&3$E04PT@<W1A('IT96UP
M(#M"1517145.H"0S,*!!3D2@)#,X#0T@:FUP(&UA:6X@.V%23U5.1*!!3D2@
M05)/54Y$H%=%H$=/+BXN#0TJ+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM
M+2TM+0TJ*J!R3U1!5$4LH%!23TI%0U0LH$%.1*!35$]21:!42$6@4$])3E13
M#2H-*J!T2$E3H$=56:!324U03%F@1T]%4Z!$3U=.H%1(1:!03TE.5*!,25-4
M+*!23U1!5$E.1Z!!3D0-*J!04D]*14-424Y'H%!/24Y44RZ@H&E4H$Y%1413
MH'B@4$%34T5$H$E.H$%3H$%.H$E.251)04P-*J!)3D1%6*!)3E1/H%1(1:!,
M25-4H$]&H%!/24Y44RR@04Y$H$E4H$%,4T^@3D5%1%.@00TJH$-/54Y415*@
M;G5M=&]R;W2@4T54H%50H$%(14%$H$]&H%1)344NH*!I5*!/3DQ9H%!23TI%
M0U13#2J@0:!#15)404E.H$Y534)%4J!/1J!03TE.5%.@4T^@5$A!5*!$249&
M15)%3E2@3T)*14-44PTJH$U!6:!"1:!-04Y)4%5,051%1*!$249&15)%3E1,
M62X-*@TJH&E4H$9)4E-4H%)/5$%415.@*#)D*:!)3J!42$6@5RU8H%!,04Y%
M+*!42$5.H%@M62R@5$A%3@TJH%DM6BR@04Y$H$9)3D%,3%F@6"U:+@TJH&E4
MH%1(14Z@4%)/2D5#5%.@5$A23U5'2*!42$6@3U))1TE.+*!&25)35*!)3E1/
MH%1(10TJH$A94$524$Q!3D6@5SU#3TY35"R@5$A%3J!)3E1/H%1(1:!03$%.
M1:!:/4-/3E-4+@TJH'1(14Z@5$A%H%!/24Y44Z!!4D6@4U1/4D5H%3H$%.
MH%@M6:!,25-4H$]&H$-/3U)$4RP-*J!7251(H$%.H$E.5$52345$24%41:!,
M25-4H$]&H%@M62U:H$-/3U)$4RR@4T^@00TJH$A)1$1%3J!&04-%H$Y/4DU!
M3*!#04Z@0D6@14%324Q9H$E-4$Q%345.5$5$+@TJ#2J@;D]41:!42$%4H&F@
M1$].)U2@5T]24EF@04)/552@1T545$E.1Z!324=.4Z!224=(5*!/4@TJH%1(
M24Y'H$Q)2T6@5$A!5*`M+:!IH$1/3B=4H$U)3D2@0:!2149,14-4142@4T],
M551)3TXN#2H-*J!N3U1%H%1(052@4$])3E13H$]224=)3D%41:!)3J!O8FQI
M<W2@04YH5.1*!54*!)3@TJH'!L:7-T+@TJ#2J@;4E.2:!354)23U5424Y%
M.J!43Z!3059%H%-/346@4U!!0T4LH%1(1:!23U1!5$E/3@TJH%!!4E2@25.@
M4U5"0T].5%)!0U1%1*!/552@5$^@04Y/5$A%4J!,25143$6@4D]55$E.12P-
M*J!R;W0N#0UR;W1P<F]J#2!L9&$@;V)L:7-T=RQX#2!S=&$@=&5M<#$-(&QD
M82!O8FQI<W1X+'@-(&QD>2!S=W@-(&IS<B!R;W0@.V&@0T].5$%)3E.@=RR@
M5$5-4#&@>`T@<W1A('!L:7-T=RQX#2!L9&$@;V)L:7-T>2QX(#MN15A4+*!8
M60T@;&1Y('-X>0T@:G-R(')O=`T@<W1A('!L:7-T>"QX#2!L9&$@;V)L:7-T
M>BQX(#MU3D2@5DA9+5I%10T@;&1Y('-Y>@T@:G-R(')O=`T@<W1A('!L:7-T
M>2QX#2!L9'D@=&5M<#$@.U1%35`QH$A!4Z!23U1!5$5$H%H-(&QD82!P;&ES
M='@L>"`[;D5%1*!23U1!5$5$H%B@3D]7#2!S=&$@=&5M<#$-('1Y82`[8T]/
M4D1)3D%415.@1$^@3D]4H$-/34U55$4A#2!L9'D@<WAZ#2!J<W(@<F]T(#MN
M3U>@6%H-('-T82!P;&ES='@L>`T@;&1A('1E;7`Q#2!S=&$@<&QI<W1Z+'@@
M.VY/5Z!%5D5264].1:!32$]53$2@0D6@4D]4051%1`T-(&QD82!P;&ES='<L
M>"`[<T\LH$Q%5"=3H$=%5*!04D]*14-424Y'(0T@8VQC#2!A9&,@(S0X(#MD
M3TXG5*!&3U)'152@5$^@4TA)1E2@6B$-('1A>0T@;&1A('!L:7-T>"QX(#MF
M25)35*!)3E1/H%<]0T].4U0-(&-L8PT@861C("-H:7-I;BLT.`T@<W1A('!R
M;VIT86(K,0T@;&1A("AP<F]J=&%B*2QY#2!S=&$@<&QI<W1X+'@-(&QD82!P
M;&ES='DL>`T@8VQC#2!A9&,@(VAI<VEN*S0X#2!S=&$@<')O:G1A8BLQ#2!L
M9&$@*'!R;VIT86(I+'D-('-T82!P;&ES='DL>`T@;&1A('!L:7-T>BQX#2!C
M;&,-(&%D8R`C:&ES:6XK-#@-('-T82!P<F]J=&%B*S$-(&QD82`H<')O:G1A
M8BDL>0T@<W1A('!L:7-T>BQX#0T@8VQC#2!A9&,@(S0X#2!T87D@.VY/5Z!)
M3E1/H%H]0T].4U0-(&QD82!P;&ES='@L>`T@8VQC#2!A9&,@(VAI<VEN*S0X
M#2!S=&$@<')O:G1A8BLQ#2!L9&$@*'!R;VIT86(I+'D-(&-L8PT@861C(&]F
M9G-E=`T@8G!L(#IO:S(@.V=505)H%'04E.4U2@8V]M<&QE=&5L>:!44D%3
M2$E.1Z!-14T-(&QD82!O9F9S970@.V1/3B=4H$=)5D6@252@0:!#2$%.0T4-
M.F]K,B!S=&$@<&QI<W1X+'@-(&QD82!P;&ES='DL>`T@8VQC#2!A9&,@(VAI
M<VEN*S0X#2!S=&$@<')O:G1A8BLQ#2!L9&$@*'!R;VIT86(I+'D-(&-L8PT@
M861C(&]F9G-E=`T@8G!L(#IO:S,-(&QD82!O9F9S970@.W!4*!)5*!43Z!4
M2$6@3U))1TE.#3IO:S,@<W1A('!L:7-T>2QX#0T@:6YX#2!D96,@;G5M=&]R
M;W0-(&)E<2`Z9&]N90T@:FUP(')O='!R;VH-.F1O;F4@<G1S#0TJ+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM+0TJH'-554)23U5424Y%H')O=`TJ
M#2J@=$A)4Z!,25143$6@1U59H$Y%1413H$%3H%-%5"U54*!414U0,3U8,:!!
M3D2@83U8,@TJH$%.1*!42$6@05!04D]04DE!5$6@04Y'3$6@3D5%1%.@5$^@
M0D6@4T54H%50H$E.H'DN#2J@=5!/3J!215154DXLH%)/5$%4142@6#*@25.@
M4U1/4D5HE.H%1%35`Q#2J@04Y$H%)/5$%4142@6#&@25.@24Z@80T-<F]T
M#2!C;&,@(#MN15A4+*!860T@861C("-H:7-I;BLT.`T@<W1A('-I;C(K,0T@
M<W1A(&-O<S(K,0T@;&1A('1E;7`Q#2!C;&,-(&%D8R`C:&ES:6XK-#@-('-T
M82!S:6XQ*S$-('-T82!C;W,Q*S$-(&QD82`H<VEN,2DL>0T@8VQC#2!A9&,@
M*&-O<S(I+'D-('-T82!T96UP,0T@;&1A("AC;W,Q*2QY#2!S96,-('-B8R`H
M<VEN,BDL>0T@<G1S#0TJ#2J@8VQE87)S970-*@TJH'1(25.@3$E45$Q%H%-5
M0E)/551)3D6@1DE,3%.@5$A%H%-#4D5%3J!7251(H%1(1:!#3TY414Y44PTJ
MH$]&H&&@04Y$H%!,04-%4Z!!H#$R6#$RH$-(05)!0U1%4J!"3$]#2Z!)3J!4
M2$6@34E$1$Q%#2J@3T:@5$A%H%-#4D5%3BR@24Z@041$251)3TZ@5$^@0TQ%
M05))3D>@5$A%H%-#4D5%3J!!3D0-*J!3151424Y'H%1(1:!"04-+1U)/54Y$
MH$-/3$]24RR@151#+@TJ#6-L96%R<V5T#2!P:&$-(&QD82`C)#`R(#MR140-
M('-T82`D9#`R,2`[=$A)4Z!)4Z!$3TY%H%-/H%1(052@3TQ$15(-(&QD82`C
M,C@-(&IS<B!C:')O=70@.RA.3U2@5$^@345.5$E/3J!.15=%4BD-(&QD82`C

M,30W(#M-04-(24Y%4Z!724Q,H%-%5*!54`T@:G-R(&-H<F]U=`T@;&1A(",V
M(#M#3U)214-43%D-('-T82`D9#`R,2`[*$%.1*!42$6@1DQ!4TB@3$]/2U.@
M0T]/3*!43T\I#0T@;&1X(",P,`T@<&QA(#MC55-43TV@0TA!4D%#5$52#3IG
M>7!S>2!S=&$@<V-R;FQO8RQX#2!S=&$@<V-R;FQO8RLR-38L>`T@<W1A('-C
M<FYL;V,K-3$R+'@-('-T82!S8W)N;&]C*S<V."QX#2!I;G@-(&)N92`Z9WEP
M<WD-#2!L9&$@(S<@.WE%3$Q/5PT@<W1A(&-O;&]R(#MT4EF@3U54H%1(052@
M4TY!6EI9H$Y%5Z!23U5424Y%#2!L9&$@(S$R#2!S=&$@;G5M<VED92`[,3*@
M0TA!4E.@4$52H%-)1$4-*J!L9&&@(S`PH#M35$%25*!#2$%2#2!T>&$-('-T
M82!S=&%R8VAA<@T@;&1X(",Q-"`[0T],#2!L9'D@(S4@.U)/5PT@:FUP('-P
M;&%T(#MS<&QA=*!724Q,H%)%5%523J!&3U*@55,-#2HM+2TM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!AH$1%4U1254-4259%H#A8.*`M/J`Q
M-J!"252@355,5$E03%F@4D]55$E.10TJ#2J@86-C*F%U>*`M/J!;86-C+$%#
M0U5-54Q!5$]27:`H3$]7+$A)*:`Q-J!"252@4D5354Q4#2J@875XH%)%34%)
M3E.@54Y#2$%.1T5$#0UM=6QT(&-L8PT@;&1A(",P,`T@;&1X(",D,#@-.FQO
M;W`@<F]R#2!R;W(@86-C#2!B8V,@.F-O;G0-(&-L8PT@861C(&%U>`TZ8V]N
M="!D97@-(&)P;"`Z;&]O<`TJH'-T8:!A=7@-(')T<PT-*BTM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM+2T-*@TJH&=E=')A;F0-*@TJH&=%3D52051%
MH$&@4T]-15=(052@4D%.1$]-H%)%4$5!5$E.1Z!315%514Y#12Z@H&F@55-%
M#2J@0:!465!)0T%,H$Q)3D5!4J!#3TY'4E5%3E1)04R@04Q'3U))5$A-#2J@
MH*"@H*!I*$XK,2F@/:`H:2A.*2I!H"N@0RF@34]HT-*J!7251(H$T]-C4U
M,S8LH$$]-2R@04YH,],3,X-#&@*"0S-C$Q*2Z@H$.@5T%3H$-(3U-%3@TJ
MH%1/H$)%H$&@4%))346@3E5-0D52H$Y%05*@*#$O,J`MH#$O-J!345)4*#,I
M*2I-+@TJ#2J@;D]41:!42$%4H$E.H$=%3D5204R@5$A%H$A)1TA%4J!"2513
MH$%21:`B34]21:!204Y$3TTB#2J@5$A!3J!42$6@3$]715*@0DE44RR@4T^@
M1D]2H$E.4U1!3D-%H$E.H%1(25.@4%)/1U)!30TJH%-)3D-%H$].3%F@4TU!
M3$R@24Y414=%4E.@*#`N+C$U+*`P+BXS.2R@151#+BF@05)%H$1%4TE2140L
M#2J@5$A%6:!32$]53$2@0D6@5$%+14Z@1E)/3:!42$6@2$E'2*!"651%H')A
M;F1O;2LQ+*!72$E#2`TJH$E3H%)%5%523D5$H$E.H&$N#2H-9V5T<F%N9`T@
M;&1A(')A;F1O;2LQ#2!S=&$@=&5M<#$-(&QD82!R86YD;VT-(&%S;`T@<F]L
M('1E;7`Q#2!A<VP-(')O;"!T96UP,0TJH&%S;`TJH')O;*!T96UP,0TJH&%S
M;`TJH')O;*!T96UP,0T@8VQC#2!A9&,@<F%N9&]M#2!P:&$-(&QD82!T96UP
M,0T@861C(')A;F1O;2LQ#2!S=&$@<F%N9&]M*S$-('!L80T@861C(",D,3$-
M('-T82!R86YD;VT-(&QD82!R86YD;VTK,0T@861C(",D,S8-('-T82!R86YD
M;VTK,0T@<G1S#0TJ+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+0TJ
M#2J@=$A)4Z!)4Z!33TU%H$-54U1/3:!#2$%204-415*@1$%403N@5$A%4T4-
M*J!715)%H%-43TQ%3J!&4D]-H'!/3%E'3TY!35DN#2H-8VAA<F1A=`UC<F]S
M<W-M(&AE>"`U-65E-35B8C4U964U-6)B#61I=&AE<C$@:&5X(&%A-35A834U
M86$U-6%A-34->FEG<R!H97@@965D9&)B-S=E961D8F(W-PUZ86=S(&AE>"`W
M-V)B9&1E93<W8F)D9&5E#6TK,2R@5TA)0T@-*J!)4Z!215154DY%1*!)3J!A
M+@TJ#6=E=')A;F0-(&QD82!R86YD;VTK,0T@<W1A('1E;7`Q#2!L9&$@<F%N
M9&]M#2!A<VP-(')O;"!T96UP,0T@87-L#2!R;VP@=&5M<#$-*J!A<VP-*J!R
M;VR@=&5M<#$-*J!A<VP-*J!R;VR@=&5M<#$-(&-L8PT@861C`'`J+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM+0TJ#2J@1U)!4$A)0U,N4PTJ#2J@
M=$A)4Z!)4Z!!H%-%5*!/1J!'4D%02$E#4Z!23U5424Y%4Z!&3U*@1$E--"R@
M1D]2H%5310TJH%=)5$B@;4523$E.H#$R."=3H$Q)3DM%4BX-*@TJH&-54E)%
M3E1,6:!)5*!#3TY404E.4Z!42$6@3$E.1:!$4D%724Y'H%)/551)3D4LH%1(
M10TJH$9)3$R@4D]55$E.12R@04Y$H$&@4D]55$E.1:!43Z!03$%#1:!"3$]#
M2U.@3T8-*J!'4D%02$E#4Z!#2$%204-415)3H$].H%1(1:!30U)%14XN#2H-
M*J!H25-43U)9.@TJH*!S;&J@H*"@-B\Y+SDVH*`M<U!,252@24Y43Z!214Q/
M0T%404),1:!,24Y+15*@1DE,10TJH*"@H*"@H*"@H*"@H*"@H*"@H*`M:4U03$5-
M14Y4142@34]$249)142@1DE,3*!23U5424Y%#2J@H*"@H*"@H*"@H*`V+S$PH*"@
MH"UP552@3TQ$H%)/551)3D6@0D%#2Z!)3J`Z*0TJH*"@H*"@H*"@H*"@H*"@
MH*`M<U150TN@24Z@34]$54Q%H%1/H$=%3D52051%H%-934U%5%))0Z!/0E,-
M*J"@H*"@H*"@H#8O,36@H*"@+69)6$5$H%50H%-934U%5%)9H%-/H%1(052@
M2$6@55-%4PTJH*"@H*"@H*"@H*"@4$%45$523E.@04Y$H%-50T@N
M#2J@H*"@H*"@H*`V+S$WH*"@H"UF25A%1*!54*!S<&QA="R@14Y(04Y#142@
M<WEM;:!0051415).4PTJH*"@H*"@H*"@-B\R,:"@H*`M9DE8142@55"@<W!L
M872@5$^@2T5%4*!&4D]-H%=2251)3D>@5$^@8VEA(0TJH*"@H*"@H*"@-B\R
M,J"@H*`M841$142@1$]45$5$H$Q)3D6@0T%004))3$E460T-(&1S:R`B1U)!
M4$A)0U,BH#MA4U-%34),1:!43Z!$25-+#0T@<F5L(#MR96Q/0T%404),10T-
M*J!C3TY35$%.5%,-#6)U9F8Q(&5Q=2`D,S`P,"`[;$]#051)3TY3H$]&H$12
M05=)3D>@0E5&1D524PUB=69F,B!E<74@)#,X,#`-8G5F9F5R,2!E<74@8G5F
M9C$K)#`T.#`@.V]&1E-%5*!43Z!314-/3D1!4EF@3T)*14-44PUB=69F97(R
M(&5Q=2!B=69F,BLD,#0X,`T-<&%T=&5R;C$@97%U("1C,3`P(#ML25-4H$]&
MH$9)3$R@4$%45$523@UP871T97)N,B!E<74@)&,Q.#`@.RA3152@55"@0EF@
M0T%,3$E.1Z!23U5424Y%*0UB:71P(&5Q=2`D8S@P,"`[8DE4H%1!0DQ%H$9/
M4J!,24Y%H%)/551)3D4-;&\T,"!E<74@)&,Y,#`@.W1!0DQ%4Z!/1J`T,"I8
M+S@-:&DT,"!E<74@)&,Y.#`@.W1/H$Q/0T%41:!/1D93152@24Y43Z!"549&
M15(-;&\Y-B!E<74@)&-A,#`@.T9/4J!!H$=)5D5.H$-/3%5-3@UH:3DV(&5Q
M=2`D8V$X,`UR979L:7-T(&5Q=2`D8V(P,"`[;$E35*!/1J!2149,14-4142@
M6`T-<V-R;FQO8R!E<74@,3`R-"`[=TA%4D6@25.@5$A%H%-#4D5%3J!,3T-!
M5$5$/PT-*J!S3TU%H%9!4DE!0DQ%4PT-8V]L<W1E<"`]("0U,2`[;$E.15,Z
MH&A/5Z!-04Y9H$)95$53H%!%4J!#3TP-8V]U;G1P=',@/2`D-3(-#6YU;6]B
M,2!E<74@)&$U(#MN54U"15*@3T:@4$])3E13H$E.H$]"2D5#5*`Q#6YU;6]B
M,3(@97%U("1A-@UT;W1N=6T@97%U("1A-R`[=$]404R@3E5-0D52H$]&H%!/
M24Y44Z!)3J!,25-4#6]F9G-E="!E<74@)&$X(#MO1D93152@24Y43Z!'4DE$
M#0UD;W1T960@97%U("1B,2`[9DQ!1Z!&3U*@1$]45$5HQ)3D6@*#$]1$]4
M5$5$*0T-<&]I;G0Q(&5Q=2`D-F$@.W-/346@4U!!4D6@4$])3E1%4E,-<&]I
M;G0R(&5Q=2`D-F,-<V-R965N<"!E<74@)#9E(#MP3TE.5$52H%53142@0EF@
M4T-2145.H%!,3U1415(-8V]L;W)P(&5Q=2`D-S`@.U!/24Y415*@24Y43Z!#
M3TQ/4J!R86T-8V]L;W(@97%U("0W,B`[8T],3U*@5$^@4U1/4D6@24Z@0T],

M3U*@<F%M#6YU;7-I9&4@97%U("0X8B`[(Z!/1J!#2$%24Z!015*@4TE$1:!/
M1J!3455!4D4-#6YU;71O<F]T(&5Q=2`D-CD@.VY534)%4J!/1J!03TE.5%.@
M5$^@4D]4051%H$E.H')O='!R;VH-#6)U9F9E<B!E<74@)&$S(#MP3TE.5$52
MH%1/H$1205=)3D>@0E5&1D52#7@Q(&5Q=2`D9F(@.W!/24Y44Z!&3U*@1%)!
M5TE.1Z!!H$Q)3D4->3$@97%U("1F8R`[=$A%4T6@6D523Z!004=%H$%$1%)%
M4U-%4PUX,B!E<74@)&9D(#M$3TXG5*!#3TY&3$E#5*!7251(H&)A<VEC#7DR
M(&5Q=2`D9F4-;VQD>"!E<74@)&9D#6-H=6YK(&5Q=2`D9F4-9'@@97%U("0V
M-PUD>2!E<74@)#8X#71E;7`Q(&5Q=2`D9F(@.V]&H$-/55)312R@0T]53$2@
M0T].1DQ)0U2@5TE42*!8,0UT96UP,B!E<74@)&9C(#MT14U03U)!4EF@5D%2
M24%"3$53#7IT96UP(&5Q=2`D,#(@.W53142@1D]2H$)51D9%4J!35T%0+J"@
M9$].)U2@5$]50T@N#6%C8R!E<74@)#(R(#MU4T5H)9H$U!5$B@4D]55$E.
M10UA=7@@97%U("0R-`UE>'0@97%U("0R-@T-*BTM+2TM+2TM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2T-*@TJH'-934U415)9H%)/551)3D4-*@TJH'1(25.@
M4D]55$E.1:!&24Q,4Z!/552@5$A%H%1/4*!&3U52H$]"2D5#5%.@0EF@0U)%
M051)3D<-*J!42$6@9#2@4UE-34544EDM4D5,051%1*!/0DI%0U13H$923TV@
M5$A%H%-%0T].1$%26:!/0DI%0U0L#2J@1D]2H$&@5$]404R@3T:@1D]54J!/
M0DI%0U13+@T-<WEM;2!E;G0-#2!L9'@@(S,Y#2!L9'D@(S`P#2!L9&$@>G1E
M;7`-(&-M<"`C/F)U9F8Q#2!B97$@.F)U9C$-(&IM<"`Z8G5F,@T-.F)U9C$@
M.V9)4E-4H$&@4D5&3$5#5$E/3J!42%)/54=(H%D],`T@;&1A(&)U9F9E<C$L
M>2`[9U)!0J!#3TQ534Y3+*!/3D6@052@0:!424U%#2!E;W(@(R1F9@T@;W)A
M('!A='1E<FXQ+'D-('-T82!B=69F97(Q*S(P,"QX#2!L9&$@8G5F9F5R,2LT
M,"QY(#MC3TR@,@T@96]R(",D9F8-(&]R82!P871T97)N,2QY#2!S=&$@8G5F
M9F5R,2LR-#`L>`T@;&1A(&)U9F9E<C$K.#`L>2`[8T],H#,-(&5O<B`C)&9F
M#2!O<F$@<&%T=&5R;C$L>0T@<W1A(&)U9F9E<C$K,C@P+'@-(&QD82!B=69F
M97(Q*S$R,"QY(#MY3U6@1T54H%1(1:!024-455)%#2!E;W(@(R1F9@T@;W)A
M('!A='1E<FXQ+'D-('-T82!B=69F97(Q*S,R,"QX#2!L9&$@8G5F9F5R,2LQ
M-C`L>0T@96]R(",D9F8-(&]R82!P871T97)N,2QY#2!S=&$@8G5F9F5R,2LS
M-C`L>`T@:6YY#2!D97@-(&)P;"`Z8G5F,0T-(&QD>2`C,SD@.VY/5Z!$3Z!2
M149,14-424].4Z!42%)/54=(H%@],`TZ;#$@;&1A(&)U9F9E<C$L>0T@=&%X
M#2!L9&$@<F5V;&ES="QX#2!A;F0@<&%T=&5R;C$L>0T@<W1A(&)U9F9E<C$K
M-38P+'D@.V-/3%5-3C&@+3Z@0T],54U.H#$P#2!L9&$@8G5F9F5R,2LT,"QY
M#2!T87@-(&QD82!R979L:7-T+'@-(&%N9"!P871T97)N,2QY#2!S=&$@8G5F
M9F5R,2LU,C`L>2`[,J`M/J`Y#2!L9&$@8G5F9F5R,2LX,"QY#2!T87@-(&QD
M82!R979L:7-T+'@-(&%N9"!P871T97)N,2QY#2!S=&$@8G5F9F5R,2LT.#`L
M>0T@;&1A(&)U9F9E<C$K,3(P+'D-('1A>`T@;&1A(')E=FQI<W0L>`T@86YD
M('!A='1E<FXQ+'D-('-T82!B=69F97(Q*S0T,"QY#2!L9&$@8G5F9F5R,2LQ
M-C`L>2`[-:`M/J`Q#2!T87@-(&QD82!R979L:7-T+'@-(&%N9"!P871T97)N
M,2QY#2!S=&$@8G5F9F5R,2LT,#`L>0T-(&QD82!B=69F97(Q*S(P,"QY(#MN
M3U>@1$^@252@1D]2H%D],*!2149,14-424].#2!T87@-(&QD82!R979L:7-T
M+'@-(&%N9"!P871T97)N,BQY#2!S=&$@8G5F9F5R,2LW-C`L>2`[8T],54U.
M,:`M/J!#3TQ534Z@,3`-(&QD82!B=69F97(Q*S(T,"QY#2!T87@-(&QD82!R
M979L:7-T+'@-(&%N9"!P871T97)N,BQY#2!S=&$@8G5F9F5R,2LW,C`L>2`[
M,J`M/J`Y#2!L9&$@8G5F9F5R,2LR.#`L>0T@=&%X#2!L9&$@<F5V;&ES="QX
M#2!A;F0@<&%T=&5R;C(L>0T@<W1A(&)U9F9E<C$K-C@P+'D-(&QD82!B=69F
M97(Q*S,R,"QY#2!T87@-(&QD82!R979L:7-T+'@-(&%N9"!P871T97)N,BQY
M#2!S=&$@8G5F9F5R,2LV-#`L>0T@;&1A(&)U9F9E<C$K,S8P+'D@.S6@+3Z@
M,0T@=&%X#2!L9&$@<F5V;&ES="QX#2!A;F0@<&%T=&5R;C(L>0T@<W1A(&)U
M9F9E<C$K-C`P+'D-#2!D97D-(&)M:2`Z9&]N93$-(&IM<"`Z;#$-.F1O;F4Q
M(')T<PT-.F)U9C(@.W-!346@5$A)3D<LH$9/4J!"549&15*@,@T@;&1A(&)U
M9F9E<C(L>2`[9U)!0J!#3TQ534Y3+*!/3D6@052@0:!424U%#2!E;W(@(R1F
M9@T@;W)A('!A='1E<FXQ+'D-('-T82!B=69F97(R*S(P,"QX#2!L9&$@8G5F
M9F5R,BLT,"QY(#MC3TR@,@T@96]R(",D9F8-(&]R82!P871T97)N,2QY#2!S
M=&$@8G5F9F5R,BLR-#`L>`T@;&1A(&)U9F9E<C(K.#`L>2`[8T],H#,-(&5O
M<B`C)&9F#2!O<F$@<&%T=&5R;C$L>0T@<W1A(&)U9F9E<C(K,C@P+'@-(&QD
M82!B=69F97(R*S$R,"QY(#MY3U6@1T54H%1(1:!024-455)%#2!E;W(@(R1F
M9@T@;W)A('!A='1E<FXQ+'D-('-T82!B=69F97(R*S,R,"QX#2!L9&$@8G5F
M9F5R,BLQ-C`L>0T@96]R(",D9F8-(&]R82!P871T97)N,2QY#2!S=&$@8G5F
M9F5R,BLS-C`L>`T@:6YY#2!D97@-(&)P;"`Z8G5F,@T-(&QD>2`C,SD@.VY/
M5Z!$3Z!2149,14-424].4Z!42%)/54=(H%@],`TZ;#(@;&1A(&)U9F9E<C(L
M>0T@=&%X#2!L9&$@<F5V;&ES="QX#2!A;F0@<&%T=&5R;C$L>0T@<W1A(&)U
M9F9E<C(K-38P+'D@.V-/3%5-3C&@+3Z@0T],54U.H#$P#2!L9&$@8G5F9F5R
M,BLT,"QY#2!T87@-(&QD82!R979L:7-T+'@-(&%N9"!P871T97)N,2QY#2!S
M=&$@8G5F9F5R,BLU,C`L>2`[,J`M/J`Y#2!L9&$@8G5F9F5R,BLX,"QY#2!T
M87@-(&QD82!R979L:7-T+'@-(&%N9"!P871T97)N,2QY#2!S=&$@8G5F9F5R
M,BLT.#`L>0T@;&1A(&)U9F9E<C(K,3(P+'D-('1A>`T@;&1A(')E=FQI<W0L
M>`T@86YD('!A='1E<FXQ+'D-('-T82!B=69F97(R*S0T,"QY#2!L9&$@8G5F
M9F5R,BLQ-C`L>2`[-:`M/J`Q#2!T87@-(&QD82!R979L:7-T+'@-(&%N9"!P
M871T97)N,2QY#2!S=&$@8G5F9F5R,BLT,#`L>0T-(&QD82!B=69F97(R*S(P
M,"QY(#MN3U>@1$^@252@1D]2H%D],*!2149,14-424].#2!T87@-(&QD82!R
M979L:7-T+'@-(&%N9"!P871T97)N,BQY#2!S=&$@8G5F9F5R,BLW-C`L>2`[
M8T],54U.,:`M/J!#3TQ534Z@,3`-(&QD82!B=69F97(R*S(T,"QY#2!T87@-
M(&QD82!R979L:7-T+'@-(&%N9"!P871T97)N,BQY#2!S=&$@8G5F9F5R,BLW
M,C`L>2`[,J`M/J`Y#2!L9&$@8G5F9F5R,BLR.#`L>0T@=&%X#2!L9&$@<F5V
M;&ES="QX#2!A;F0@<&%T=&5R;C(L>0T@<W1A(&)U9F9E<C(K-C@P+'D-(&QD
M82!B=69F97(R*S,R,"QY#2!T87@-(&QD82!R979L:7-T+'@-(&%N9"!P871T
M97)N,BQY#2!S=&$@8G5F9F5R,BLV-#`L>0T@;&1A(&)U9F9E<C(K,S8P+'D@
M.S6@+3Z@,0T@=&%X#2!L9&$@<F5V;&ES="QX#2!A;F0@<&%T=&5R;C(L>0T@
M<W1A(&)U9F9E<C(K-C`P+'D-#2!D97D-(&)M:2`Z9&]N93(-(&IM<"`Z;#(-
M.F1O;F4R(')T<PT-#2HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM
M#2H-*J!F24Q,H%)/551)3D4-*@TJH'1(25.@25.@5$A%H$],1*!#3T]+244M
M0U545$52+*!34$E&1D5$H%50H$%.1*!!H$))5`TJH$]05$E-25I%1"R@4TE.

M0T6@252@25.@3TY,6:!54T5H].H%1(1:!&25)35*`U6#6@1U))1`TJ#2J@
M9FEL;&QI<W2@+2V@0:!(04Y$6:!404),1:!43Z!414Q,H%=(24-(H%)/5U.@
M2$%61:!"145.#2J@H*"@H*"@H*"@1DE,3$5$+@T-9FEL;"!E;G0-#2J@
M;&1YH",S.0TJH&QD8:`C,#`-*CIL,:!S=&&@9FEL;&QI<W0L>0TJH&1E>0TJ
MH&)P;*`Z;#$-#2!L9'D@(S,Y#3IL;V]P,2!L9&$@(R0X,"`[;D5%1*!43Z!!
M1$2@)#`T.#"@5$^@1T54H%1/#2!S=&$@8G5F9F5R(#M&25)35*`U6#6@1U))
M1`T@;&1A('IT96UP#2!C;&,-(&%D8R`C)#`T#2!S=&$@8G5F9F5R*S$-#2J@
M=$A%H$],1*!705DN#2J@;$]44Z!/1J!!1$13H$%.1*!354-(+*!"552@4T5%
M35.@5$^@0D6@0:!"252@1D%35$52#0T@;&1X(",P-0TZ;&]O<#(@;&1A("AB
M=69F97(I+'D-(&)N92`Z9V]T8VAA#2!L9&$@(R1F9B`[9DE,3*!&3$%4+*!0
M051415).4Z!,051%4@T@<W1A("AB=69F97(I+'D-(&QD82!B=69F97(-(&%D
M8R`C-#`-('-T82!B=69F97(-(&)C8R`Z8V]N=`T@:6YC(&)U9F9E<BLQ#2!C
M;&,-.F-O;G0@9&5X#2!B;F4@.FQO;W`R#2!B97$@.FYO=&AI;F<-#3IG;W1C
M:&$@;&1X(",D9F8-.FPQ(&EN>`T@87-L(#MB250LH$))5"R@5TA/)U.@1T]4
MH%1(1:!"250_#2!B8V,@.FPQ#0T@;&1A(&)I='`L>`T@96]R(",D9F8-(&]R
M82`H8G5F9F5R*2QY#2!S=&$@*&)U9F9E<BDL>0T-*J!N3U>@5T]22Z!"04-+
M5T%21%,N#2J@<D5-14U"15(ZH"0P-#@PH"N@,38PH#V@)#`U,C`-(&QD82`C
M)#(P#2!S=&$@8G5F9F5R#2!L9&$@>G1E;7`-(&-L8PT@861C(",D,#4-('-T
M82!B=69F97(K,0T-.FPT('-E8PTZ;&]O<#,@;&1A("AB=69F97(I+'D-(&)N
M92`Z9&EN9PT@;&1A(",D9F:@.V9)3$R@1DQ!5"R@4$%45$523E.@3$%415(-
M('-T82`H8G5F9F5R*2QY#2!L9&&@(&)U9F9E<@T@<V)C(",T,`T@<W1A(&)U
M9F9E<@T@8F-S(#IL;V]P,Z`[8D]53D2@5$^@2$E4H%-/34542$E.1Z!33T].
M15(-(&1E8R!B=69F97(K,:`[3U*@3$%415(N#2!B8V,@.FPT#0TZ9&EN9R!L
M9'@@(R0P.*`[=T6@2$%61:!!H%=)3DY%4B$-.FPR(&1E>`T@;'-R#2!B8V,@
M.FPR#2!L9&&@(&)I='`L>`T@;W)A("AB=69F97(I+'D-('-T82`H8G5F9F5R
M*2QY#0TZ;F]T:&EN9Z!D97D-(&)P;"`Z;&]O<#&@.VM%15"@1T])3D>@54Y4
M24R@>:!(2513H%I%4D\-#2J@=T5,3"R@04Y/5$A%4J!03U-324))3$E462R@
M0E54H$U!64)%H$Y/5*!!4Z!'3T]$#2H-*J!I5*!'3T53H%1(4D]51TB@5$A%
MH$Q)4U2@5D525$E#04Q,62R@5$A54Z!32TE04$E.1PTJH$%,3*!42$]31:!#
M3TQ534XM041$4RZ@H'1(1:!$3U=.4TE$1:!)4Z!!3$R@5$A%#2J@0T]-4$%2
M25-/3E,NH*!I5*!!4U-53453H%1(052@052@3$5!4U2@3TY%H$Q)3D4-*J`H
M12Y'+J!9/3(P*:!724Q,H$Y%142@5$^@0D6@0T]-4$Q%5$5,6:!&24Q,142@
M1T])3D<-*J!&3U)705)$4RX-*@TJ.FQO;W`RH&QD>:"@(S,YH#MF3U)705)$
M4Z!005)4#2HZ;&]O<#&@;&1AH&9I;&QL:7-T+'D-*J!B;F6@.G-K:7`-*J!L
M9&&@*&)U9F9E<BDL>0TJH&)E<:`Z;F]T>65T#2J@<W1AH&9I;&QL:7-T+'F@
M.VY/3EI%4D^@24U03$E%4Z!&24Q,140-*J!T87B@.V]/4%,LH%1)346@5$^@
M1DE,3`TJH&QD8:"@8FET<"QX#2HZ;F]T>65TH&5O<J`C)&9F#2J@<W1AH"AB
M=69F97(I+'D-*CIS:VEPH&1E>0TJH&)P;*`Z;&]O<#$-*J!L9&&@8G5F9F5R
M#2J@8G!LH#IB86-KH#LD6%@X,*`KH#$V,*`]H%A8,#"@*Z`S,@TJH&-L8PTJ
MH&%D8Z`C-#`-*J!S=&&@8G5F9F5R#2J@8F-CH#IL;V]P,@TJH&EN8Z!B=69F
M97(K,0TJH&)C<Z`Z;&]O<#(-*@TJ.F)A8VN@.VY/5Z!73U)+H$)!0TM705)$
M4PTJ.FQO;W`SH'-E8Z`[=$A)4Z!,1513H%53H%%5252@14%23%DLH$].0T4-
M*J`[159%4EE/3D6@25.@1DE,3$5$+@TJH&QD>:`C,SF@.V9/4E=!4D13H%!!
M4E0-*CIL;V]P-*!L9&&@9FEL;&QI<W0L>0TJH&)E<:`Z<VMI<#(-*J!C;&.@
M.V9,04>@24:@04Y9H$5.5%)9H$Y/3EI%4D\-*J!L9&&@*&)U9F9E<BDL>0TJ
MH&)E<:`Z;F]T979E;@TJH'1A>*`[;T]04RR@5$E-1:!43Z!&24Q,#2J@;&1A
MH*`C,#`-*J!S=&&@9FEL;&QI<W0L>:`[>D523Z!/552@1DE,3*!&3$%'4PTJ
MH'-E8PTJH&QD8:!B:71P+'@-*CIN;W1E=F5N#2J@<W1AH"AB=69F97(I+'D-
M*CIS:VEP,J!D97D-*J!B<&R@.FQO;W`T#2J@8F-SH#ID;VYEH#MW1:!#04Z@
M1DE.25-(H$5!4DQ9#2J@;&1AH&)U9F9E<@TJH'-E8PTJH'-B8Z`C-#`-*J!S
M=&&@8G5F9F5R#2J@8G!LH#ID;VYEH#LD6%@X,*!)4Z!!4Z!,3U>@05.@252@
M1T]%4RX-*J!B8W.@.FQO;W`S#2J@9&5CH&)U9F9E<BLQ#2J@8F-CH#IL;V]P
M,PTJ.F1O;F4-(')T<PT-#2HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM
M+2TM#2J@9%)!5TE.)Z!!H$Q)3D4NH*!AH$9!2$Z@3$%(3BX-#2HJ*J!S3TU%
MH%5314953*!-04-23U,-#6-I;FET(&UA8R`@.VU!0U)/H%1/H$E.251)04Q)
M6D6@5$A%H$-/54Y415(-(&QD82!=,2`[1%B@3U*@1%D-(&QS<@T@/#P\("`[
M=$A%H$18+S*@34%+15.@0:!.24-%4J!,3T]+24Y'H$Q)3D4-#2HJ*BHJH&U!
M0U)/H%1/H%1!2T6@0:!35$50H$E.H'@-#7AS=&5P(&UA8PT@;&1X(&1X(#MN
M54U"15*@3T:@3$]/4*!)5$52051)3TY3#2`^/CX@8VEN:70L9'@->&QO;W`@
M;'-R(&-H=6YK#2!B97$@9FEX8R`[=5!$051%H$-/3%5-3@T@<V)C(&1Y#2!B
M8V,@9FEX>2`[=$E-1:!43Z!35$50H$E.H'D-8V]N="!D97@-(&)N92!X;&]O
M<`UD;VYE(&QD82!C:'5N:R`[<$Q/5*!42$6@3$%35*!#2%5.2PT@;'-R(#MM
M04M%H%-54D6@5$^@1T54H$%4H$Q%05-4H#&@4$E814P-(&5O<B!O;&1X#2!A
M;F0@9&]T=&5D#2!O<F$@*&)U9F9E<BDL>0T@<W1A("AB=69F97(I+'D-(')T
M<PT-9FEX8R!P:&$-(&QD82!O;&1X#2!A;F0@9&]T=&5D(#M$3U14142@3$E.
M1:!-05-+#2!O<F$@*&)U9F9E<BDL>2`[<$Q/5`T@<W1A("AB=69F97(I+'D-
M(&QD82`C)&9F(#MU4$1!5$6@0TA53DL-('-T82!O;&1X#2!S=&$@8VAU;FL-
M(&QD82!B=69F97(@.VE.0U)%05-%H%1(1:!#3TQ534X-(&-L8PT@861C(&-O
M;'-T97`-('-T82!B=69F97(-(&)C8R!C,@T@:6YC(&)U9F9E<BLQ#6,R#2!P
M;&$-('-E8PT@<V)C(&1Y#2!B8W,@8V]N="`[=5"@04)/5D4-(&%D8R!D>`T@
M:68@:2Q=,2`[9$^@5T6@55-%H&EN>:!/4J!D97D_#2!I;GD-(&5L<V4-(&1E
M>0T@9FEN#2!B8W,@8V]N="`[84Q705E3H$)204Y#2$53#2IC;VYTH&1E>`TJ
MH&)N9:!X;&]O<`TJH&IM<*!D;VYE#0UF:7AY(&%D8R!D>`T@<&AA#2!L9&$@
M;VQD>`T@96]R(&-H=6YK#2!A;F0@9&]T=&5D#2!O<F$@*&)U9F9E<BDL>0T@
M<W1A("AB=69F97(I+'D-(&QD82!C:'5N:PT@<W1A(&]L9'@-('!L80T@:68@
M:2Q=,2`[=5!$051%H'D-(&EN>0T@96QS90T@9&5Y#2!F:6X-(&)C<R!C;VYT
M(#MA9&.@4T544Z!42$6@0T%24EF@04)/5D4-(#P\/"`@.V5.1*!/1J!M04-2
M3Z!84U1%4`T-*BHJ*BJ@=$%+1:!!H%-415"@24Z@>0T->7-T97`@;6%C#2!L
M9'@@9'D@.VY534)%4J!/1J!,3T]0H$E415)!5$E/3E,-(&)E<2!D;VYE(#MI
M1J!$63TPH$E4)U.@2E535*!!H%!/24Y4#2`^/CX@8VEN:70L9'D-('-E8PUY
M;&]O<"!P:&$-('1Y82`[;TY,6:!03$]4H$].H$5614Z@3$E.15.@248-(&%N

M9"!D;W1T960@.T1/24Y'H%1(1:!$3U14142@5$A)3D<N#2!B;F4@<VMI<&ET
M(#LH1$]45$5$/3`QH$E&H%=%H$%21:!$3U1424Y'*0T@;&1A(&]L9'@-(&]R
M82`H8G5F9F5R*2QY#2!S=&$@*&)U9F9E<BDL>0US:VEP:70@<&QA#2!I9B!I
M+%TQ#2!I;GD-(&5L<V4-(&1E>0T@9FEN#2!S8F,@9'@-(&)C8R!F:7AX#6-O
M;G0@9&5X#2!B;F4@>6QO;W`-9&]N92!L9&$@;VQD>`T@;W)A("AB=69F97(I
M+'D-('-T82`H8G5F9F5R*2QY#2!R=',-#69I>'@@861C(&1Y#2!L<W(@;VQD
M>`T@<V5C("`[:4U03U)404Y4(0T@8FYE(&-O;G0@.U50#2J@8F5QH&9I>&,-
M*J!D97@-*J!B;F6@>6QO;W`-*J!J;7"@9&]N90T-9FEX8R!P:&$-(')O<B!O
M;&1X(#MC05)26:!)4Z!3152@*%1224-+62$I#2!L9&$@8V]L<W1E<`T@861C
M(&)U9F9E<@T@<W1A(&)U9F9E<@T@8F-C(&,R#2!I;F,@8G5F9F5R*S$-8S(@
M<&QA#2!S96,@(#MC05)26:!.145$4Z!43Z!"1:!3150-(&)C<R!C;VYT#2J@
M9&5X#2J@8FYEH'EL;V]P#2J@:FUPH&1O;F4-(#P\/"`@.V5.1*!/1J!M04-2
M3Z!94U1%4`T-*BHJ*J!I3DE424%,H$Q)3D6@4T5455`-#2J@8V]L<W1E<*`M
M+:!H3U>@34%.6:!"651%4Z!015*@0T],54U.+*!%251(15*@-#"@3U*@.38-
M*J!L;S0P+*!H:30P+*!%5$,NH"TMH'1!0DQ%4Z!/1J`T,"I8+S@-*J!X,2QX
M,BQY,2QY,J`M+:!03TE.5%.@5$^@1%)!5Z!,24Y%H$)%5%=%14X-*J!B=69F
M97*@+2V@>G"@4$])3E1%4J!43Z!"549&15*@5$^@1%)!5Z!)3E1/#2J@>G1E
M;7"@+2V@8T].5$%)3E.@2$E'2*!"651%H$]&H$)51D9%4J!"05-%#2J@9&]T
M=&5DH"TMH'-%5*!43Z`D,#&@24:@1$])3D>@1$]45$5$H$1205=3H"A$24Q)
M1T5.5$Q9*0T-9')A=R!E;G0-#3IS971U<"!S96,@(#MM04M%H%-54D6@6#$\
M6#(-(&QD82!X,@T@<V)C('@Q#2!B<&P@.F-O;G0-(&QD82!Y,B`[:4:@3D]4
M+*!35T%0H'`QH$%.1*!P,@T@;&1Y('DQ#2!S=&$@>3$-('-T>2!Y,@T@;&1A
M('@Q#2!L9'D@>#(-('-T>2!X,0T@<W1A('@R#0T@<V5C#2!S8F,@>#$@.VY/
M5Z!A/418#3IC;VYT('-T82!D>`T@;&1X('@Q(#MP552@6#&@24Y43Z!X+*!.
M3U>@5T6@0T%.H%1205-(H'@Q#0UC;VQU;6X@;&1A(&-O;'-T97`@.V9)3D2@
M5$A%H$9)4E-4H$-/3%5-3J!&3U*@>`T@8VUP(",Y-B`[84Y$H%-%5*!54*!4
M2$6@1%)!5TE.1Z!"549&15)3#2!B97$@.F-O;G0Y-@T@;&1A(&QO-#`L>"`[
M:4:@-#`LH%1(14Z@5T6@3D5%1*!43Z!/1D93152@24Y43PTJH&-L8PT@861C
M(",D.#`@.U1(1:!"549&15*@0EF@."HQ,BHQ,CTD-#@P#2!S=&$@8G5F9F5R
M#2!L9&$@:&DT,"QX#2!A9&,@(R0P-"`[8T]53$2@159%3J!)3D-/4E!/4D%4
M1:!)3E1/H%1!0DQ%#2!B8V,@.F1O;F4-#3IC;VYT.38@;&1A(&QO.38L>"`[
M;U1(15)725-%H%=%H%-405)4H$%4H$)%1TE.3DE.1PT@<W1A(&)U9F9E<B`[
M3T:@0E5&1D52#2!L9&$@:&DY-BQX#2!C;&,-.F1O;F4@861C('IT96UP#2!S
M=&$@8G5F9F5R*S$-#2!S96,-(&QD82!Y,B`[8T%,0U5,051%H$19#2!S8F,@
M>3$-(&)P;"`Z8V]N=#(@.VE3H%DR/EDQ/PT@96]R(",D9F8@.V]42$525TE3
M1:!$63U9,2U9,@T@861C(",D,#$@.V-!4E)9H'-H;W5L9*!"1:!#3$5!4@TZ
M8V]N=#(@<W1A(&1Y#2!C;7`@9'@@.W=(3R=3H$))1T=%4CJ@1%F@3U*@1%@_
M#2!B8V,@<W1E<&EN>"`[:4:@1%@LH%1(14XN+BX-(&IM<"!S=&5P:6YY#0US
M=&5P:6YX(&QD82!D;W1T960-(&)E<2`Z8V]N=`T@;&1A(",E,3`Q,#$P,3`@
M.V1/5%1%1*!,24Y%H$U!4TL-.F-O;G0@96]R(",D9F8-('-T82!D;W1T960-
M(&QD>2!Y,0T@8W!Y('DR(#M9,2Q9,BQ/3$18+$-(54Y+H%-(05)%H%-!346@
M345-#2!L9&$@8FET<"QX(#MXH$-54E)%3E1,6:!#3TY404E.4Z!8,0T@<W1A
M(&]L9'@-('-T82!C:'5N:PT@8F-S('AD96-Y(#MD3Z!71:!35$50H$9/4E=!
M4D13H$]2H$)!0TM705)$4Z!)3J!Y/PT->&EN8WD@/CX^('AS=&5P+&EN>0T-
M>&1E8WD@/CX^('AS=&5P+&1E>0T-<W1E<&EN>2!L9'D@>3$@.VY/5$6@:::!!
M4U-5346@9&]T=&5DH#V@,#"@3U*@,#$-(&QD82!B:71P+'@@.W@]6#$-('-T
M82!O;&1X#2!L<W(@(#MYH$1/15-.)U2@55-%H$-(54Y+4PT@96]R(&]L9'@@
M.W-/H%=%H$I54U2@5T%.5*!42$6@0DE4#2!S=&$@;VQD>`T@8W!Y('DR#2!B
M8W,@>61E8WD-#7EI;F-Y(#X^/B!Y<W1E<"QI;GD-#7ED96-Y(#X^/B!Y<W1E
M<"QD97D-#2HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!T
M2$E3H%)/551)3D6@5TE,3*!03$%#1:!!H%-154%21:!#2%5.2Z!/1J!#2$%2
M04-415)3#2J@052@0:!34$5#249)142@4$Q!0T6@24Z@345-3U)9+*!!3D2@
M4U1/4D4-*J!#3TQ/4J!)3J!42$6@0T]24D534$].1$E.1Z!#3TQ/4J!-14U/
M4EDN#2H-*J!O3J!%3E1263J@>3U23U<LH'@]0T],54U.+*!A/7-405)424Y'
MH$-(05)!0U1%4@TJH*"@H*"@H*"@H*!C;VQO<CU#3TQ/4J"@;G5M<VED93TC
MH$]&H$-(05)!0U1%4E.@4$52#2J@H*"@H*"@H*"@H*"@H*"@H*"@H*"@H*"@
MH*"@H*"@H%-)1$6@3T:@4U%505)%#0US<&QA="!E;G0-('-T82!T96UP,0T@
M;&1A(",P,`T@<W1A('-C<F5E;G`K,0T@='EA(#MN3U>@5T6@3D5%1*!43Z`J
M-#`-(&%S;"`[5TA)0TB@25.@*C,RH"N@*C@-(&%S;`T@87-L#2!S=&$@=&5M
M<#(@.U1)3453H#B@*$Y/H%=/4E-%H%1(04Z@,3DV*0T@87-L#2!R;VP@<V-R
M965N<"LQ#3IC;VYT,2!A<VP-(')O;"!S8W)E96YP*S$-.F-O;G0R(&%D8R!T
M96UP,@T@<W1A('-C<F5E;G`-('-T82!C;VQO<G`-(&QD82!S8W)E96YP*S$-
M(&%D8R`C,#0@.W-#4D5%3J!35$%25%.@052@)#`T,#`-('-T82!S8W)E96YP
M*S$-(&%D8R`C)&0T(#M#3TQ/4J!!5*`D9#@P,`T@<W1A(&-O;&]R<"LQ(#MW
M3U<LH$Q/5%.@3T:@0T]$1:!42$521:`Z*`T-('-T>"!T96UP,@T@;&1A(&YU
M;7-I9&4-('-T82!A=7@@.VY%142@5$A)4Z!'55F@5$]/H#HH#3IL,B!L9&$@
M=&5M<#$-(&QD>"!N=6US:61E#2!L9'D@=&5M<#(-.FPQ('-T82`H<V-R965N
M<"DL>0T@<&AA#2!L9&$@8V]L;W(-('-T82`H8V]L;W)P*2QY#2!P;&$-(&EN
M>0T@861C(&YU;7-I9&4@.V-(05)!0U1%4E.@05)%H$]21$52142@5D525$E#
M04Q,60T@9&5X#2!B;F4@.FPQ#2!I;F,@=&5M<#$-(&QD82!S8W)E96YP#2!C
M;&,-(&%D8R`C-#`-('-T82!S8W)E96YP#2!S=&$@8V]L;W)P#2!B8V,@.F-O
M;G0T#2!I;F,@<V-R965N<"LQ#2!I;F,@8V]L;W)P*S$-.F-O;G0T(&QD82!C
M;VQO<G`K,0T@8VUP(",D9&(@.V1/3B=4H$=%5*!43T^@0DE'(2$A#2!B8V,@
M.F-O;G0U#2!L9&$@8V]L;W)P#2!C;7`@(R1C,2`[=T4G4D6@4$%35*!42$6@
M3$%35*!23U<LH$%.1`T@8F-S(#ID;VYE(#MIH$1/3B=4H$9%14R@3$E+1:!7
M4DE424Y'H%1/H&-I80TZ8V]N=#4@9&5C(&%U>`T@8FYE(#IL,@T-.F1O;F4@
M<G1S(#MG3T]$H&Q/4D2@::!(3U!%H%1(050G4Z!)5"X-ZSU=DJS;.OF2KNL^
MW;JG)79ZLV:Y*O.[K+DJ2;SM.\K/L^WKJEF3;Y5V2KJLFRU=DJS;.LN2K3M6
MS;5O)79*O.[K+DJ2;3M6S;5O*ZI9DV^5=DJZK-LM79*MFWKY5V2K+LZTY*MZ
M[[=O.V;.L^2I[M-R7+?)RW3>YSWR?/2G+?)S]*=5\G.<]\G/<[2GV<]\GN=E
MTW+<US7Y5V2K.L^Z5=DJTK>L^2IZLV:U=DJV;>M.2KNL^W;JFM79*LVSMZ4I

M2@!P*BTM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2T-*@TJH$]"2D5#
M5%,N4PTJ#2J@=$A%4T6@05)%H%1(1:!23U5424Y%4Z!72$E#2*!3152@55"@
M5$A%H$]"2D5#5%.@1D]2#2J@5$A%H$U!24Z@4%)/1U)!32Z@H&E4H$E3H$%,
M5T%94Z!!4U-5345$H%1(052@5$A%#2J@5RU#3T]21$E.051%H$]&H%1(1:!&
M25)35*!03TE.5*!/1J!!3EF@3T)*14-4#2J@5TE,3*!%5D5.5%5!3$Q9H$)%
MH$].1:`H,C2@04-454%,3%DI+*!33Z!42$6@0T%,3$E.1PTJH%)/551)3D6@
M5TE,3*!+3D]7H%=(14Z@5$A%H$]"2D5#5*!(05.@0D5%3J!#3TU03$5414Q9
M#2J@1$5&24Y%1*`H04Y$H%-50E-%455%3E1,6:!32TE0H%1(15-%H%)/551)
M3D53*2X-*@TJH'1/H")'4D]7(J!!3J!/0DI%0U2@5$A%4T6@1U594Z!324U0
M3%F@4U1!4E2@0:!#3T]21$E.051%#2J@052@6D523Z!!3D2@1U)!1%5!3$Q9
MH%)!35"@252@55"@5$^@,C0N#2H-*J!S5$502$5.H&PNH&I51$0-*J`V+S$U
M+SDV#2H-*J!H25-43U)9.@TJH*"@H*"@H*`V+S$UH"US4$Q)5*!)3E1/H$]"
M2D5#5*!&24Q%+*!#3$5!3D5$H%50H$&@1D57H%1(24Y'4PTJH*"@H*"@H*`V
M+S(QH"UA1$1%1*!/0DI%0U13H$%.1*!33D5!2UDM4$541:!$149)3DE424].
MH$%,1UD-*J"@H*"@H*"@-B\R,J`M841$142@1$]45$5$H$Q)3D6@0T%004))
M3$E462Z@H'1/H$U!2T6@5$A)3D=3#2J@H*"@H*"@H*"@H*"@H$U/4D6@0T]-
M4%)%2$5.4TE"3$4LH$Q)3D53H$-/3DY%0U1)3D>@(DE.3D52(@TJH*"@H*"@
MH*"@H*"@H*!!3D2@(D]55$52(J!03U)424].4RR@12Y'+J!03TE.5%.@5TE4
M2*!7/3$-*J"@H*"@H*"@H*"@5$^@4$])3E13H%=)5$B@5STM,2R@05)%
MH$1205=.H$1/5%1%1"Z@H'-%10TJH*"@H*"@H*"@H*!/0DI%0U2@,Z!$
M149)3DE424].H$Q)4U2@1D]2H$U/4D6@24Y&3RX-*J"@H*"@H*"@-B\R,Z`M
M841$142@5$A%H%)%34%)3DE.1Z!/0DI%0U13#0T@9'-K(")/0DI%0U13(@T-
M(')E;`T-*J!C3TY35$%.5%,-#6)U9F8Q(&5Q=2`D,S`P,"`[9DE24U2@0TA!
M4D%#5$52H%-%5`UB=69F,B!E<74@)#,X,#`@.W-%0T].1*!#2$%204-415*@
M4T54#0US:6YT86(@97%U("1C,#`P(#MT04),1:!/1J!324Y%4PUH:7-I;B!E
M<74@)#8P(#MS24Z@5$%"3$6@0U524D5.5$Q9H%-405)44Z!!5*`D-C`P,`UO
M8FQI<W1W(&5Q=2`D8S0P,"`[=$A%H$%#5%5!3*!/0DI%0U2@4$])3E13#6]B
M;&ES='@@97%U("1C-#@P#6]B;&ES='D@97%U("1C-3`P#6]B;&ES='H@97%U
M("1C-3@P#7!L:7-T=R!E<74@)&,V,#`@.VQ)4U2@3T:@4D]4051%1"M04D]*
M14-4142@4$])3E13#7!L:7-T>"!E<74@)&,V.#`-<&QI<W1Y(&5Q=2`D8S<P
M,`UP;&ES='H@97%U("1C-S@P#6)I='`@97%U("1C.#`P(#MB252@5$%"3$6@
M1D]2H$Q)3D6@4D]55$E.10UL;S0P(&5Q=2`D8SDP,"`[=$%"3$53H$]&H#0P
M*E@O.`UH:30P(&5Q=2`D8SDX,"`[=$^@3$]#051%H$]&1E-%5*!)3E1/H$)5
M1D9%4@UL;SDV(&5Q=2`D8V$P,"`[1D]2H$&@1TE614Z@0T],54U.#6AI.38@
M97%U("1C83@P#7)E=FQI<W0@97%U("1C8C`P(#MU4T5H)9H%-934U%5%)9
MH%)/551)3D4-8V]N;&ES=#$@97%U("1C8S`P(#MF25)35*!03TE.5*!)3J!#
M3TY.14-424].H$Q)4U0-8V]N;&ES=#(@97%U("1C9#`P(#MS14-/3D2@4$])
M3E2@24Z@0T].3D5#5$E/3J!,25-4#61R87=C;VX@97%U("1C93`P(#ML25-4
MH$]&H$-/3DY%0U1)3TZ@24Y$24-%4PUC;VYF;&%G(&5Q=2`D8V8P,"`[9DQ!
M1U.@5$^@2T5%4*!&4D]-H%)%1%)!5TE.1Z!35%5&1@T-<V-R;FQO8R!E<74@
M,3`R-"`[=TA%4D6@25.@5$A%H%-#4D5%3J!,3T-!5$5$/PT-*J!S3TU%H%9!
M4DE!0DQ%4PT-8V]L<W1E<"`]("0U,2`[;$E.15,ZH&A/5Z!-04Y9H$)95$53
MH%!%4J!#3TP-8V]U;G1P=',@/2`D-3(-=WAI;F,@/2`D-3,->'EI;F,@/2`D
M-30->7II;F,@/2`D-34->'II;F,@/2`D-38-<W=X(#T@)#8S#7-X>2`]("0V
M-"`[=$A%4T6@05)%H%1(1:!!3D=,15.@55-%1*!"60US>7H@/2`D-C4@.W)O
M='!R;VH-<WAZ(#T@)#8V#0UN=6UO8C$@97%U("1A-2`[;E5-0D52H$]&H%!/
M24Y44Z!)3J!/0DI%0U2@,0UN=6UO8C$R(&5Q=2`D838-=&]T;G5M(&5Q=2`D
M83<@.W1/5$%,H$Y534)%4J!/1J!03TE.5%.@24Z@3$E35`UO9F9S970@97%U
M("1A."`[;T9&4T54H$E.5$^@1U))1`UN=6UC;VYS(&5Q=2`D83D@.W1/5$%,
MH$Y534)%4J!/1J!#3TY.14-424].4PT-:6UO;F]B(&5Q=2`D8C`@.W=(24-(
MH$]"2D5#5*!!3:!IH$]./PT-<VEN,2!E<74@)#4W(#MP3TE.5$524Z!43Z!3
M24Y%H$%.1*!#3U-)3D4-8V]S,2!E<74@)#4Y(#M404),15,N#7-I;C(@97%U
M("0U8@UC;W,R(&5Q=2`D-60-<')O:G1A8B!E<74@)#5F(#MP3TE.5$52H%1/
MH%!23TI%0U1)3TZ@5$%"3$53#7!O:6YT,2!E<74@)#9A(#MS3TU%H%-005)%
MH%!/24Y415)3#7!O:6YT,B!E<74@)#9C#7-C<F5E;G`@97%U("0V92`[<$])
M3E1%4J!54T5H)9H%-#4D5%3J!03$]45$52#6-O;&]R<"!E<74@)#<P(#M0
M3TE.5$52H$E.5$^@0T],3U*@<F%M#6-O;&]R(&5Q=2`D-S(@.V-/3$]2H%1/
MH%-43U)%H$E.H$-/3$]2H')A;0UN=6US:61E(&5Q=2`D.&(@.R.@3T:@0TA!
M4E.@4$52H%-)1$6@3T:@4U%505)%#0UN=6UT;W)O="!E<74@)#8Y(#MN54U"
M15*@3T:@4$])3E13H%1/H%)/5$%41:!)3J!R;W1P<F]J#0UB=69F97(@97%U
M("1A,R`[<$])3E1%4J!43Z!$4D%724Y'H$)51D9%4@UX,2!E<74@)&9B(#MP
M3TE.5%.@1D]2H$1205=)3D>@0:!,24Y%#7DQ(&5Q=2`D9F,@.W1(15-%H%I%
M4D^@4$%'1:!!1$1215-315,->#(@97%U("1F9"`[1$].)U2@0T].1DQ)0U2@
M5TE42*!B87-I8PUY,B!E<74@)&9E#6]L9'@@97%U("1F9`UC:'5N:R!E<74@
M)&9E#61X(&5Q=2`D-C<-9'D@97%U("0V.`UT96UP,2!E<74@)&9B(#MO1J!#
M3U524T4LH$-/54Q$H$-/3D9,24-4H%=)5$B@6#$-=&5M<#(@97%U("1F8R`[
M=$5-4$]205)9H%9!4DE!0DQ%4PUZ=&5M<"!E<74@)#`R(#MU4T5$H$9/4J!"
M549&15*@4U=!4"Z@H&1/3B=4H%1/54-(+@UA8V,@97%U("0R,B`[=5-%1*!"
M6:!-051(H%)/551)3D4-875X(&5Q=2`D,C0-97AT(&5Q=2`D,C8-#2HM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!S971C;VYS#2H-*J!T
M2$E3H%!23T-%1%521:!31513H%50H$-/3DY%0U1)3TZ@3$E35%,-*@TJH&E.
M4%544SH-*J"@H*!N=6UC;VYSH$-/3E1!24Y3H%1/5$%,H$Y534)%4J!/1J!#
M3TY.14-424].4PTJH*"@H'!L:7-T=Z!#3TY404E.4Z!42$6@0T].3D5#5$E/
M3J!,25-4+*!%+D<NH&]B,6-O;@TJH*"@H'!L:7-T>:!#3TY404E.4Z!42$6@
M3D]234%,+4-/3DY%0U1)3TZ@3$E35*!%+D<NH&YO<FTQ#2H-*J!I5*!214%,
M3%F@25.@1D%35$52H$%.1*!32$]25$52H%1/H$I54U2@0T]06:!42$6@3$E3
M5%,N#2J@;TZ@15A)5"R@8V]N;&ES=#$LH&-O;FQI<W0R+*!!3D2@9')A=V-O
M;J!!4D6@4T54H%50+@TJ#0US971C;VYS#2!L9'@@(S`P(#MN3U>@4T54H%50
MH$-/3DY%0U1)3TY3#2!L9'D@(S`P#3IL,R!L9&$@<&QI<W1W+'D@.V9)4E-4
MH%!/24Y4#2!S=&$@8V]N;&ES=#$L>`T@:6YY#2!L9&$@<&QI<W1W+'D@.W-%
M0T].1*!03TE.5`T@<W1A(&-O;FQI<W0R+'@-(&EN>0T@:6YX#2!C<'@@;G5M

M8V]N<R`[=$]404R@3E5-0D52H$]&H$-/3DY%0U1)3TY3#2!B;F4@.FPS#2`[
M9DE.04Q,62R@4T54H%50H%1(1:!.3U)-04R@0T].3D5#5$E/3E,-(&QD>"`C
M,#`-.FPT('1X82`[*'-405)4H%=)5$B@0:!,25-4H$]&H%1(1:`T9*!#3TY3
M*0T@<W1A(&1R87=C;VXL>`T@:6YX#2!C<'@@<&QI<W1Y(#MN54U"15*@3T:@
M-&2@0T].3D5#5$E/3E,-(&)N92`Z;#0-(&QD>2`C,#`@.VY/5Z!&24Y)4TB@
M55"@5$A%H$Q)4U0-.FPU(&QD82!P;&ES='DK,BQY#2!S=&$@9')A=V-O;BQX
M#2!I;G@-(&EN>0T@8W!Y('!L:7-T>2LQ(#MT2$6@3E5-0D52H$]&H%!/24Y4
M4Z!)3J!42$6@3$E35`T@8FYE(#IL-0T@<G1S#0TJ+2TM+2TM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+0TJ#2J@9W)O=PTJ#2J@=$A)4Z!354)23U5424Y%
MH")'4D]74R*@5$A%H%!/24Y44RR@0EF@4D%-4$E.1Z!42$5-#2J@3U545T%2
M1%.@1E)/3:!42$6@3U))1TE.+J"@8EF@55-%H$]&H$&@34%32Z!)5*!)4PTJ
MH%!/4U-)0DQ%H%1/H$=23U>@4U5"4T544Z!/1J!03TE.5%.@052@0:!424U%
M+@TJ#2J@;TZ@14Y44EDLH'F@0T].5$%)3E.@5$A%H$Y534)%4J!/1J!03TE.
M5%.@5$^@1U)/5RTQH"@P+BYY*2P-*J!P;VEN=#&@25.@4T54H%50H$%3H$&@
M4$])3E1%4J!43Z!42$6@4$])3E2@3$E35"P-*J!!3D2@8:!#3TY404E.4Z!!
MH$U!4TLN#2H-*J!T2$6@4$])3E2@3$E35*!73U)+4Z!!4Z!&3TQ,3U=3.J!4
M2$521:!!4D6@0:!43U1!3*!/1@TJH$9/55*@0T]/4D1)3D%415.@1D]2H$5!
M0TB@5D525$58+*!!3D2@14%#2*!#3T]21$E.051%H$E3#2J@14E42$52H#`L
MH#LH]2H"TQ+J"@=$A54Z!71:!#04Z@4D504D5314Y4H$5!0TB@0T]/4D1)
M3D%410TJH$)9H%173Z!"2513+*!!3D2@5$A%H953R@5D5#5$]2H$U!6:!"
M1:!215!215-%3E1%1*!"6:!!#2J@4TE.1TQ%H$)95$4NH*!L152@,#`],"R@
M,#$],2R@04Y$H#$Q/2TQ+*!42$5.H$%.H$5.5%)9#2J@3$E+10TJH*"@H*"@
MH#`Q,3`P,#`PH*`]H"0V,`TJH%=/54QH-/4E)%4U!/3D2@5$^@0:!615)4
M15B@052@*%<L6"Q9+%HI/2@Q+"TQ+#`L,"DN#2H-*J!T2$6@4U5"4D]55$E.
M1:!404M%4Z!42$E3H$Y534)%4BR@86YD4Z!)5*!7251(H%1(1:!-05-+#2J@
M4U1/4D5HE.H'1E;7`Q+*!!3D2@4$%24T53H%1(1:!215-53%1)3D>@5D%,
M544NH*!I1@TJH$&@0T]/4D1)3D%41:!)4Z`Q+*!42$5.H%1(1:!#3U)215-0
M3TY$24Y'H$-/3U)$24Y!5$6@24X-*J!42$6@4$])3E2@3$E35*`H;V)L:7-T
M=RR@151#+BF@25.@24Y#4D5-14Y4140NH*!I1@TJH$&@0T]/4D1)3D%41:!)
M4Z`M,2R@;V)L:7-T=Z!)4Z!$14-214U%3E1%1"Z@H&E&H#`LH%1(14X-*J!)
M5*!)4Z!)1TY/4D5$+J"@:4Z@5$A)4Z!705DLH$E.1$E6241504R@4$])3E13
MH$U!6:!"10TJH$=23U=.H$]55%=!4D13+*!&4D]-H#"@5$^@,C0NH*!B6:!5
M4TE.1Z!42$6@34%32Z!)5*!)4PTJH%!/4U-)0DQ%H%1/H$=23U<LH%-!62R@
M3TY,6:!42$6@6"U#3T]21$E.051%4RR@3U*@3TY,60TJH%!/4TE4259%H$-/
M3U)$24Y!5$53+@TJ#6=R;W<-('-T82!T96UP,0TZ;&]O<"!T>6$-('1A>`T@
M;&1A("AP;VEN=#$I+'D-(&%N9"!T96UP,0TZ8VAE8VMW(&%S;`T@8F-C(#IP
M;W-W#2!A<VP@.VI54U2@05-354U%H$Y%1T%4259%#2!D96,@;V)L:7-T=RQX
M#2!B;F4@.F-H96-K>`TZ<&]S=R!A<VP-(&)C8R`Z8VAE8VMX#2!I;F,@;V)L
M:7-T=RQX#3IC:&5C:W@@87-L#2!B8V,@.G!O<W@-(&%S;`T@9&5C(&]B;&ES
M='@L>`T@8FYE(#IC:&5C:WD-.G!O<W@@87-L#2!B8V,@.F-H96-K>0T@:6YC
M(&]B;&ES='@L>`TZ8VAE8VMY(&%S;`T@8F-C(#IP;W-Y#2!A<VP-(&1E8R!O
M8FQI<W1Y+'@-(&)N92`Z8VAE8VMZ#3IP;W-Y(&%S;`T@8F-C(#IC:&5C:WH-
M(&EN8R!O8FQI<W1Y+'@-.F-H96-K>B!A<VP-(&)C8R`Z<&]S>@T@87-L#2!D
M96,@;V)L:7-T>BQX#2!B;F4@.F1O;F4-.G!O<WH@87-L#2!B8V,@.F1O;F4-
M(&EN8R!O8FQI<W1Z+'@-#3ID;VYE(&1E>0T@8G!L(#IL;V]P#2!R=',-#0TJ
M+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+0TJ#2J@9$5&24Y%H$]"
M2D5#5%,-*@TJH'1(25.@3D585*!04D]#14154D6@25.@5$A%H%!23U1/5%E0
M24-!3*!04D]#14154D4-*J!&3U*@4T545$E.1Z!54*!!3D2@1$5&24Y)3D>@
M0:!005)424-53$%2H$]"2D5#5*!'4D]54"X-*J!I5*!$3T53H%-/H$)9H")'
M4D]724Y'(J!42$6@4$])3E13H$%.1*!3151424Y'H$%.1TQ%4PTJH$%04%)/
M4%))051%3%DNH*!W2$5.H$1/3D4LH%1(15)%H$E3H$&@0T]-4$Q%5$6@4$])
M3E0-*J!,25-4H%1/H$)%H%)/5$%4140LH$540RX-#2J@;G5M;V(QH*"@+2V@
M;E5-0D52H$]&H%!/24Y44Z!)3J!/0DI%0U2@3TY%#2J@=&]T;G5MH*"@+2V@
M=$]404R@3E5-0D52H$]&H%!/24Y44Z!43Z!"1:!23U104D]*)T0-#61E9F]B
M<R!E;G0-(&QD82!I;6]N;V(-(&)N92`Z8VAE8VLQ#2!J;7`@9&5F;V(P#3IC
M:&5C:S$@8VUP(",Q#2!B;F4@.F-H96-K,@T@:FUP(&1E9F]B,0TZ8VAE8VLR
M(&-M<"`C,@T@8FYE(#IC:&5C:S,-(&IM<"!D969O8C(-.F-H96-K,PT-*@TJ
MH&1E9F]B,Z`M+:!S152@55"@3T)*14-4H",S+*!!H$-50D4-*@UD969O8C,-
M.FEN:70-#2!L9&$@(S,P(#MS25A4145.H#1D+*!%24=(5*`S9"M325B@3D]2
M34%,4PT@8VUP('1O=&YU;0T@8F5Q(#IG<F]W(#MO3DQ9H%)53J!42$6@24Y)
M5*!23U5424Y%H$].0T4-#2!S=&$@=&]T;G5M#2!L9&$@(S$V#2!S=&$@;G5M
M;V(Q(#MP4DE-05)9H"@T9"D-(&QD82`C,30-('-T82!N=6UO8C$R(#MS14-/
M3D1!4EF@*#-D*:`KH$Y/4DU!3%,-(&QD82`C-#0-('-T82!N=6UC;VYS(#LT
M-*!#3TY.14-424].4Z`H,"XN-#,I#0T@;&1A(",D9C@@.W1(25.@2T5%4%.@
M5$A%H#-DH$]"2D5#5*!&4D]-#2!L9'@@(S@@.T=%5%1)3D>@5$]/H$))1Z`H
M4TY%04M9+*!%2"D-.FPQ('-T82!O8FQI<W1W*S$U+'@-(&1E>`T@8FYE(#IL
M,0T-*J!S152@55"@5$A%H$-/3DY%0U1)3TZ@3$E35%,-*F]B,6-O;J`M/J!P
M;&ES='<LH#0TH%!44PTJ;F]R;3&@+3Z@<&QI<W1Y+*`T,*!05%.@5$]404P-
M(&QD>2`C.#<@.S0T*C*@4$])3E13H%!%4J!#3TY.14-424].#3IL;V]P(&QD
M82!O8C-C;VXL>0T@<W1A('!L:7-T=RQY#2!L9&$@;F]R;3,L>2`[>45!2"R@
M4T^@5T6@0T]06:!!H$)53D-(H$]&#2!S=&$@<&QI<W1Y+'D@.T585%)!H$I5
M3DLLH$))1Z!72$]/4"X-(&1E>0T@8G!L(#IL;V]P#0T@:G-R('-E=&-O;G,-
M*@TJH&F@05-354U%H$Y/4DU!3%.@2$%61:!"145.H$-,14%2142@3U54H$)9
MH$-!3$Q)3D>@4D]55$E.10TJ#2!L9&$@(S(T(#MN3U>@4T54H%50H%1(1:!.
M3U)-04Q3#2!S=&$@;V)L:7-T>"LR-"`[*R\MH#$L,"PPH$540RZ@1D]2H$&@
M0U5"10T@<W1A(&]B;&ES='DK,C8-('-T82!O8FQI<W1Z*S(X#2!L9&$@(S(U
M-BTR-`T@<W1A(&]B;&ES='@K,C4-('-T82!O8FQI<W1Y*S(W#2!S=&$@;V)L
M:7-T>BLR.0T-.F=R;W<-(&QD82`C/'9E<G0S#2!S=&$@<&]I;G0Q#2!L9&$@
M(SYV97)T,PT@<W1A('!O:6YT,2LQ#2!L9'D@(S(S(#MN54U"15*@3T:@4$])
M3E13H%1/H$=23U>@,"XN>0T-*J!G4D]7H%1(1:!8+4-/3U)$4Z!&25)35"X-
M#3IX8V]O<F1S(&QD82!O8FQI<W1X#2!C;7`@(S(T#2!B97$@.GEC;V]R9',-

M(&QD82`C)3`P,3$P,#`P(#MM05-+#2!J;7`@9W)O=R`[9W)O=Z!724Q,H')T
M<Z!&3U*@55,N#0TZ>6-O;W)D<R!L9&$@;V)L:7-T>0T@8VUP(",R-`T@8F5Q
M(#IZ8V]O<F1S#2!L9&$@(R4P,#`P,3$P,`T@:FUP(&=R;W<-#3IZ8V]O<F1S
M(&QD82!O8FQI<W1Z(#MH059%H&F@2$E4H#(TH"A-05B@5D%,544IH%E%5#\-
M(&-M<"`C,C0-(&)E<2`Z=V-O;W)D<PT@:6YC('-X>2`[;4%+1:!42$6@3T)*
M14-4H$&@3$E45$Q%#2!I;F,@<WEZ(#M.24-%4J!,3T]+24Y'+@T@;&1A(",E
M,#`P,#`P,3$-(&IM<"!G<F]W#0TZ=V-O;W)D<R!L9&$@;V)L:7-T=PT@8VUP
M(",R,PT@8FYE(#IC;VYT#2!L9'D@(R1F9@T@<W1Y('AY:6YC#2!L9'D@(S`Q
M(#MS5$%25*!42$6@04Y'3$53H$=/24Y'#2!S='D@=WAI;F,-('-T>2!Y>FEN
M8PT@:6YY#2!S='D@>'II;F,-.F-O;G0-(&QD82`C)3$Q,#`P,#`P#2!J;7`@
M9W)O=PT-*@TJH'1(25.@25.@5$A%H$Q)4U2@3T:@5D525$E#15,LH$%3H$1%
M4T-224)%1*!)3J!G<F]WH$%"3U9%+@TJH#`PH#V@,"R@,#&@/:`Q+*`Q,:`]
MH"TQ#2H-=F5R=#,@9&9B("4P,3`Q,#$P,2`[,2PQ+#$L+3$-(&1F8B`E,#$P,3$Q,#$-(&1F8B`E,#$P,3$Q,3$-
M(&1F8B`E,#$Q,3`Q,#$-(&1F8B`E,#$Q,3`Q,3$-(&1F8B`E,#$Q,3$Q,#$-
M(&1F8B`E,#$Q,3$Q,3$-(&1F8B`E,3$P,3`Q,#$@.RTQ+#$L,2PQ#2!D9F(@
M)3$Q,#$P,3$Q#2!D9F(@)3$Q,#$Q,3`Q#2!D9F(@)3$Q,#$Q,3$Q#2!D9F(@
M)3$Q,3$P,3`Q#2!D9F(@)3$Q,3$P,3$Q#2!D9F(@)3$Q,3$Q,3`Q#2!D9F(@
M)3$Q,3$Q,3$Q(#LM,2PM,2PM,2PM,:"@14Y$H$]&H#1DH$-50D6@3$E35`T@
M9&9B("4P,#`Q,#$P,2`[,"PQ+#$L,:"@,V2@0U5"10T@9&9B("4P,#`Q,#$Q
M,0T@9&9B("4P,#`Q,3$P,0T@9&9B("4P,#`Q,3$Q,0T@9&9B("4P,#$Q,#$P
M,0T@9&9B("4P,#$Q,#$Q,0T@9&9B("4P,#$Q,3$P,0T@9&9B("4P,#$Q,3$Q
M,2`[14YH]&H#-DH$-50D6@3$E35"R@,C2@4$])3E13H%1/5$%,#0TJ#2J@
M=$A%4T6@05)%H%1(1:!#3TY.14-424].4Z!"1517145.H%!/24Y44PTJ#2J@
M<D5-14U"15*@5$A!5*!42$531:!!4D6@:6YD:6-E<Z!)3E1/H%1(1:!W:&]L
M9:!03TE.5`TJH$Q)4U0LH$Y/5*!*55-4H$&@3$]#04R@4$])3E0A(2$-*@TJ
MH'1(15-%H$-/3DY%0U1)3TY3H$%21:!!3$R@4$Q!0T5HE.5$^@0:!,25-4
MH%=(24-(H$U!60TJH$)%H$%#0T534T5$H$)9H%1(1:!-04E.H$1205=)3D>@
M4D]55$E.12Z@H'1(14Z@5$A%#2J@3D]234%,4Z!!4D6@4$Q!0T5$H$E.H%1(
M14E2H$]73J!,25-4+*!!3$].1Z!7251(#2J@24Y$24-%4Z!)3E1/H%1(1:!#
M3TY.14-424].H$Q)4U0NH*!T2$6@3D]234%,H$Q)4U2@5T]22U,-*J!!4Z!&
M3TQ,3U=3.@TJ#2J@=$A%H$9)4E-4H%-%5*!/1J!.54U"15)3H$E.1$5815.@
M1$E214-43%F@24Y43Z!42$4-*J!04DE-05)9H$]"2D5#5*!#3TY.14-424].
M4Z`M+:!42$521:!!4D6@3D^@3D]234%,4PTJH%1/H%=/4E)9H$%"3U54+*!!
M3D2@5$A)4Z!!3$Q/5U.@5$A%H$]"2D5#5%.@5$^@0D4-*J!44D%.4U!!4D5.
M5$Q9H$A!3D1,142@0EF@5$A%H$1205=)3D>@4D]55$E.12Z@H&]42$525TE3
M10TJH$5!0TB@1D%#1:!/1J!42$6@4T5#3TY$05)9H$E3H$A!3D1,142@5$A%
MH%-!344Z#2J@=$A%H$9)4E-4H$Y534)%4J!)4Z!42$6@24Y$15B@5$^@5$A%
MH$Y/4DU!3*!)3J!42$6@4$])3E0-*J!,25-4H"AP;&ES="DNH*!F3TQ,3U=)
M3D>@5$A)4Z!)3D1%6*!)4Z!!H$Q)4U2@3T:@24Y$24-%4PTJH$E.5$^@5$A%
MH$-/3DY%0U1)3TZ@3$E35*`M+:!42$531:!#3TY.14-424].4Z!-04M%H%50
MH%1(10TJH$9!0T6@5TA)0TB@0D5,3TY'4Z!43Z!42$6@3D]234%,+J"@=$A)
M4Z!#3TY.14-424].H$Q)4U0-*J!)4Z!415)-24Y!5$5H)9H$&@6D523RR@
M04Y$H$9/3$Q/5T5$H$)9H$]42$52H$Y/4DU!3`TJH$Q)4U13+J"@=$A%H%=(
M3TQ%H%1(24Y'H$E3H%1%4DU)3D%4142@5TE42*!!H%-%0T].1*!:15)/+@TJ
M#2J@=$A54Z!42$6@1%)!5TE.1Z!23U5424Y%H%=/4DM3H$%3H$9/3$Q/5U,Z
MH$-/3DY%0U1)3TY3#2J@1E)/3:!42$6@0T].3D5#5$E/3J!,25-4H$%21:!3
M54-#15-3259%3%F@4D5!1*!)3J!!3D0-*J!*3TE.140LH%5.5E,H&@6D52
M3Z!)4Z!(250NH*!T2$6@3D585*!03TE.5*!)4Z!42$5.#2J@4D5!1*!)3J!!
M4Z!!H$Y/4DU!3*!)3D1%6"Z@H&E&H%I%4D\LH%1(14Z@5T6@05)%H$1/3D4L
M#2J@3U1(15)725-%H%1(25.@3D]234%,H$E3H$-(14-+142@5$^@4T5%H$E&
MH%1(1:!&04-%#2J@25.@5DE324),12Z@H&E&H$Y/5*!42$5.H$E4H%-+25!3
MH%1/H%1(1:!.15A4H%!/24Y4+`TJH$]42$525TE31:!)5*!$3T53H%1(1:!$
M4D%724Y'H%1(24Y'H$%"3U9%+@TJ#2J@85.@14%#2*!#3TY.14-424].H$E3
MH$1205=.H$&@0T]24D534$].1$E.1Z!&3$%'H$E3#2J@4T54H$E.H$&@4TE-
M24Q!4J!#3TY.14-424].H$Q)4U0NH*!T2$E3H%=!62R@0T].3D5#5$E/3E,-
M*J!32$%2142@0D545T5%3J!625-)0DQ%H$9!0T53H$1/H$Y/5*!.145$H%1/
MH$)%H$1205=.H%1724-%+@TJ#2J@;D57H$9%05154D4ZH$Y/5Z!#3TY.14-4
M24].4Z!-05F@0D6@1%)!5TZ@55-)3D>@14E42$52#2J@4T],242@3U*@1$]4
M5$5$H$Q)3D53+J"@=$A%H$-50D6@1$]%4Z!.3U2@3D5%1*!)5"R@0E54#2J@
M252@2$5,4%.@24U-14Y314Q9H%=)5$B@5$A%H$]42$524Z!43Z!$4D%7H$Q)
M3D53H$-/3DY%0U1)3D<-*J`B24Y.15(BH$%.1*`B3U5415(BH$]"2D5#5%.@
M5TE42*!$3U14142@3$E.15,LH%1/H%-%1:!(3U<-*J!42$E.1U.@0TA!3D=%
MH%=(14Z@5$A%6:!35$%25*!43Z!455).H$E.4TE$12U/550NH*!S24Y#1:!4
M2$5210TJH$%21:!.3U2@1T])3D>@5$^@0D6@34]21:!42$%.H#$R-Z!#3TY.
M14-424].4RR@5$A%H$A)1T@-*J!"252@3T:@5$A%H$-/3DY%0U1)3TZ@24Y$
M15B@4T525D53H$%3H$&@1$]45$5HQ)3D6@1DQ!1SH-*J!)1J!42$6@2$E'
M2*!"252@3T:@5$A%H$E.1$58H$E3H%-%5"R@5$A%3J!$4D%7H%1(1:!#3TY.
M14-424Y'#2J@3$E.1:!!4Z!$3U14140NH*!H14Y#1:!42$6@1%)!5TE.1Z!2
M3U5424Y%H$].3%F@3D5%1%.@5$\-*J!35%))4*!/1D:@5$A)4Z!"252@04Y$
MH%-%5*!42$6@1DQ!1Z!D;W1T962@04-#3U)$24Y'3%DN#0UO8C-C;VX@:&5X
M(#`P,#$P,#`R,#`P-#`P,#@@.S`M,:`P+3*@,"TTH#`M.`T@:&5X(#`Q,#,P
M,3`U,#$P.2`[,2TSH#$M-:`Q+3D-(&AE>"`P,C`S,#(P-C`R,&$@.S(M,Z`R
M+3:@,BUA#2!H97@@,#,P-S`S,&(@("`[,RTWH#,M8@T@:&5X(#`T,#4P-#`V
M,#0P8R`[-"TUH#0M-J`T+6,-(&AE>"`P-3`W,#4P9"`@(#LU+3>@-2UD#2!H
M97@@,#8P-S`V,&4@("`[-BTWH#8M90T@:&5X(#`W,&8@("`[-RUF#2!H97@@
M,#@P.3`X,&$P.#!C(#LX+3F@."UAH#@M8PT@:&5X(#`Y,&(P.3!D("`@.SDM
M8J`Y+60-(&AE>"`P83!B,&$P92`@(#M!+4*@02U%#2!H97@@,&(P9B`@(#M"
M+48-(&AE>"`P8S!D,&,P92`@(#M#+42@0RU%#2!H97@@,&0P9@T@:&5X(#!E
M,&8-;V(S,F-O;B`[,V2@0U5"10T@:&5X(#$P,3$Q,#$R,3`Q-"`[,38M,3>@
M,38M,3B@,38M,C`-(&AE>"`Q,3$S,3$Q-2`[,3<M,3F@,3<M,C&@H"@S-2D-

M(&AE>"`Q,C$S,3(Q-B`[,3@M,3F@,3@M,C(-(&AE>"`Q,S$W(#LQ.2TR,Z"@
MH*"@H*"@*#,Y*0T@:&5X(#$T,34Q-#$V(#LR,"TR,:`R,"TR,@T@:&5X(#$U
M,3<@.S(Q+3(SH*"@H*"@H*`H-#(I#2!H97@@,38Q-R`[,C(M,C,-*J`T9#J@
M,S*@0T].3D5#5$E/3E,LH#-D.J`Q,J!#3TY.14-424].4PUN;W)M,R`[;$E3
M5*!/1J!.3U)-04Q3H"N@0T].3D5#5$E/3E,-(&AE>"`R,"`[9DE24U2@3E5-
M0D52H$E3H$Y534)%4J!/1J`T9*!#3TY3#2!D9F(@,S@@.W-%0T].1*!.54U"
M15*@25.@3E5-0D52H$]&H%!/24Y44PT[24Z@3$E35"R@0D5,3U<-(&AE>"`P
M,#$X(#MN3U)-04R@,:!!5*!)3D1%6*`R-*``H#$L,"PP#2!D9F(@,S(@.S$V
M+3$W#2!D9F(@,S,@.S$V+3$X#2!D9F(@,S4@.S$W+3$Y#2!D9F(@,S<@.S$X
M+3$Y#2!H97@@,#`Q.2`[;D]234%,H",RH#V@+3$L,"PP#2!D9F(@-#`@.S(P
M+3(Q#2!D9F(@-#$@.S(P+3(R#2!D9F(@-#(@.S(Q+3(S#2!D9F(@-#,@.S(R
M+3(S#2!H97@@,#`Q82`[;D]234%,H",S/:`P+#$L,`T@9&9B(#,R#2!D9F(@
M,S8-(&1F8B`T,`T@9&9B(#,T#2!H97@@,#`Q8B`[;D]234%,H#0]H#`L+3$L
M,`T@9&9B(#,W#2!D9F(@,S@-(&1F8B`S.0T@9&9B(#0S#2!H97@@,#`Q8R`[
M;D]234%,H#4]H#`L,"PQ#2!D9F(@,S,-(&1F8B`S-`T@9&9B(#0Q#2!D9F(@
M,S@-(&AE>"`P,#%D(#MN3U)-04R@-CV@,"PP+"TQ#2!D9F(@,S4-(&1F8B`S
M-@T@9&9B(#,Y#2!D9F(@-#(-(&AE>"`P,#`P(#ME3D2@3R>@5$A%H$Q)4U0-
M#0TJ+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+0TJ#2J@9&5F;V(P
MH"TMH'-%5*!54*!/0DI%0U2@(S`LH$&@5$544D%(14123TX-*@TJH'1(25.@
M25.@3TY%H$]&H%1(1:!/0DI%0U13H&F@1$52259%H$923TV@5$A%H$-50D4-
M*J!615)424-%4RX-*@TJH'9%4E1)0T53H$%21:!!3$R@(E173Z!!5T%9(J!&
M4D]-H$5!0TB@3U1(15*@3TZ@5$A%H$-50D4N#2J@<U1!4E1)3D>@1E)/3:!!
MH#-DH$-50D4LH$].1:!354-(H%-%5*!/1J!03TE.5%.@25,-*J`H,2PQ+#$I
MH"@Q+"TQ+"TQ*:`H+3$L,2PM,2F@04Y$H"@M,2PM,2PQ*:`M+:!43Z!'150-
M*J!&4D]-H$].1:!03TE.5*!43Z!42$6@3D585*!215%525)%4Z!44D%615)3
M24Y'H%173PTJH$Q)3D6@4T5'345.5%.@3TZ@5$A%H$-50D4NH*!G14Y%4D%,
M25I)3D>@5$^@-&0LH&D-*J!!1T%)3J!54T6@0:!3152@3T:@4$])3E13H$].
MH%1(1:!#54)%H%=(24-(H$1)1D9%4@TJH$E.H%173Z!#3T]21$E.051%4Z!!
M3D2@24Z@-*!#3T]21$E.051%4RR@12Y'+J!35$%25$E.1PTJH$923TV@*#$L
M,2PQ+#$IH%1(1:!45T\M05=!6:!03TE.5%.@05)%H"@Q+#$L+3$L+3$IH$54
M0RX-*J!!3D2@5$A%H$9/55(M05=!6:!03TE.5*!)4Z`H+3$L+3$L+3$L+3$I
M+@TJ#2J@:4:@64]5H%=!3E2@5$^@0D6@34]21:!34$5#249)0RR@::!42$E.
M2Z!42$E3H$9/4DU3H$$-*J!#3TU03$541:!3152@3T:@4$])3E13H$5154E6
M05))04Y4H%5.1$52H%1(1:!!0U1)3TZ@3T8-*J!42$6@4D5,159!3E2@4UE-
M34544EF@1U)/55"@*$%#5$E.1Z!/3J!/3D6@4$])3E2@5TE42`TJH$&@4UE-
M34544EF@3U!%4D%43U*@1T5.15)!5$53H$%.3U1(15*@4$])3E2@24Z@5$A%
MH%-%5"DN#2H-*J!A3D]42$52H%=!6:!/1J!,3T]+24Y'H$%4H$E4H$E3H%1(
M052@::!33$E#1:!/1D:@0T]23D524PTJH$]&H%1(1:!#54)%H%=(24-(H$%2
M14XG5*!!1$I!0T5.5*!43Z!%04-(H$]42$52+@TJ#2J@;D]41:!42$%4H$Y/
M4DU!3%.@0T]24D534$].1*!%6$%#5$Q9H%1/H%1(1:`S9*!03TE.5%,N#2J@
M=$A!5*!)4RR@3T:@04Y9H$9!0T6@1D]2345H)9H%1(4D5%H%!/24Y44RR@
M5$A%H$9/55)42`TJH$9/4DU3H$&@5D5#5$]2H$Y/4DU!3*!43Z!42$6@1D%#
M1:`H04Q"14E4H%!/24Y424Y'#2J@05=!6:!&4D]-H$E4*2X-*@T-9&5F;V(P
M#3II;FET#2!L9&$@(S$V(#LXH#1D+*`TH#-D*S2@3D]234%,4PT@8VUP('1O
M=&YU;0T@8F5Q(#IG<F]W(#MO3DQ9H%)53J!42$6@24Y)5*!23U5424Y%H$].
M0T4-#2!S=&$@=&]T;G5M#2!L<W(@.VF@3D5%1*!42$]31:!"651%4R$A(0TJ
MH&QD8:`C.`T@<W1A(&YU;6]B,2`[<%))34%26:`H-&0I#2J@;&1AH",X#2!S
M=&$@;G5M;V(Q,B`[<T5#3TY$05)9H"@S9"F@*Z`TH$Y/4DU!3%,-(&QD82`C
M,S`-('-T82!N=6UC;VYS(#LS,*!#3TY.14-424].4Z`H,"XN,CDI#0T@;&1X
M(",T#2!L9&$@(R1F."`[;4%+1:!/0DI%0U2@0:!,25143$6@4TU!3$Q%4@TZ
M;#$@<W1A(&]B;&ES='<K-RQX#2!D97@-(&)N92`Z;#$-#2J@<T54H%50H%1(
M1:!#3TY.14-424].H$Q)4U13#2IO8C!C;VZ@+3Z@<&QI<W1W+*`S,*!#3TY.
M14-424].4Z`H,J!05%.@4$52H$-/3BD-*FYO<FTPH"T^H'!L:7-T>2R@,C2@
M4%13H%1/5$%,#2!L9'D@(S4Y(#LS,"HR#3IL;V]P(&QD82!O8C!C;VXL>0T@
M<W1A('!L:7-T=RQY#2!L9&$@;F]R;3`L>0T@<W1A('!L:7-T>2QY#2!D97D-
M(&)P;"`Z;&]O<`T-(&IS<B!S971C;VYS#0TJ#2J@<T54H%=(24-(H$-/3DY%
M0U1)3TY3H%=)3$R@55-%H$1/5%1%1*!,24Y%4PTJH&1/5%1%1*!#3TY3H$%2
M13J@,"TTH#`M-:`P+3:@,2TTH#$M-:`Q+3>@,BTTH#(M-J`R+3<-*J"@H*"@
MH*"@H*"@H*"@H*"@H#,M-:`S+3:@,RTW#2!L9'D@(S$Q#3IL,B!L9&$@9&]T
M8V]N,"QY#2!T87@-(&QD82!D<F%W8V]N+'@-(&]R82`C)#@P#2!S=&$@9')A
M=V-O;BQX#2!D97D-(&)P;"`Z;#(-#3IG<F]W#2!L9&$@(SQV97)T,`T@<W1A
M('!O:6YT,0T@;&1A(",^=F5R=#`-('-T82!P;VEN=#$K,0T@;&1Y(",Q-2`[
M;E5-0D52H$]&H%!/24Y44Z!43Z!'4D]7H#`N+GD-(&QD82`C)&9F(#MG4D]7
MH$%,3*!#3T]21%.@15%504Q,60T@:FUP(&=R;W<-#2H-*J!N15A4H%1(1:!6
M15)415B@3$E35`TJ#79E<G0P(&1F8B`E,#$P,3`Q,#$@.S$L,2PQ+#$-(&1F
M8B`E,#$P,3$Q,3$-(&1F8B`E,#$Q,3`Q,3$-(&1F8B`E,#$Q,3$Q,#$-(&1F
M8B`E,3$P,3`Q,3$-(&1F8B`E,3$P,3$Q,#$-(&1F8B`E,3$Q,3`Q,#$-(&1F
M8B`E,3$Q,3$Q,3$@.V5.1*!/1J`T9*!03TE.5%,-(&1F8B`E,#`P,3`Q,#$-
M(&1F8B`E,#`P,3$Q,3$-(&1F8B`E,#`Q,3$Q,3$-(&1F8B`E,#`Q,3$Q,#$@
M.S$RH%!/24Y44Z!43U1!3`T@9&9B("4P,#$Q,3$Q,0T@9&9B("4P,#$Q,#$P
M,0T@9&9B("4P,#`Q,3$P,0T@9&9B("4P,#`Q,#$Q,2`[<$Q54Z!&3U52H$Y/
M4DU!3%,-#6]B,&-O;B!H97@@,#`P,3`P,#(P,#`S,#`P-#`P,#4P,#`V(#LP
M+3&@,"TRH#`M,Z`P+32@,"TUH#`M-@T@:&5X(#`Q,#(P,3`S,#$P-#`Q,#4P
M,3`W(#LQ+3*@,2TSH#$M-*`Q+36@,2TW#2!H97@@,#(P,S`R,#0P,C`V,#(P
M-R`[,BTSH#(M-*`R+3:@,BTW#2!H97@@,#,P-3`S,#8P,S`W(#LS+36@,RTV
MH#,M-PT@:&5X(#`T,#4P-#`V,#0P-PT@:&5X(#`U,#8P-3`W#2!H97@@,#8P
M-R`[-&2@0T].3D5#5$E/3E.@*#(T*0T-;V(P,F-O;B!H97@@,#@P.3`X,&$P
M.#!B(#LX+3F@."TQ,*`X+3$Q#2!H97@@,#DP83`Y,&(-(&AE>"`P83!B(#LS
M9*!#3TY.14-424].4Z`H-BD-#61O=&-O;C`@:&5X(#`S,#0P-2`[=$A%4T6@
M0T].3D5#5$E/3E.@5TE,3*!"1:!$3U14140-(&AE>"`P.#`Y,&$@.U1(15F@
M0T].3D5#5*!03TE.5%.@5TE42*!7/3&@5$\-(&AE>"`P8S!D,&4@.U!/24Y4

M4Z!7251(H%<]+3$-(&AE>"`P9C$P,3$@.S$RH%1/5$%,#0UN;W)M,"`[;D]2
M34%,4RM#3U)215-03TY$24Y'H$-/3DY%0U1)3TY3#2!D9F(@,C0@.VY534)%
M4J!/1J`T9*!#3TY.14-424].4PT@9&9B(#(R(#MN54U"15*@3T:@4$])3E13
MH$)%3$]7#2!H97@@,#`P8R`[;D]234%,H$%4H$E.1$58H#$RH#V@,2PQ+#$-
M(&1F8B`R-R`[.2TQ,`T@9&9B(#(X(#LY+3$Q#2!D9F(@,CD@.S$P+3$Q#2!H
M97@@,#`P9"`[;D]234%,H#$L+3$L+3$-(&1F8B`R-0T@9&9B(#(V#2!D9F(@
M,CD-(&AE>"`P,#!E(#MN3U)-04R@+3$L,2PM,0T@9&9B(#(T(#LX+3$Y#2!D
M9F(@,C8@.S@M,3$-(&1F8B`R."`[.2TQ,0T@:&5X(#`P,&8@.VY/4DU!3*`M
M,2PM,2PQ#2!D9F(@,C0@.S@M.0T@9&9B(#(U(#LY+3$P#2!D9F(@,C<@.SDM
M,3`-(&AE>"`P,#`P(#LR,J!03TE.5%.@24Z@3$E35`T-*BTM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM+2T-*@TJH&1E9F]B,@TJ#2J@<T54H%50H$]"
M2D5#5*`C,BR@04Z@3T-404A%1%)/3BX-*@TJH'=(052@24Z@5$A%H%=/4DQ$
MH$E3H%1(1:`T9*!!3D%,3TY'H$]&H$%.H$]#5$%(14123TX_#2H-*J!T2$52
M1:!!4D6@5%=/H%=!65.@3T:@3$]/2TE.1Z!!5*!42$E3+J"@8:`S9*!/0U1!
M2$5$4D].#2J@2$%3H$-/3U)$24Y!5$53H"@K+RTQ+#`L,"F@*#`L*R\M,2PP
M*:`H,"PP+"LO+3$I+*!33PTJH$].1:!03U-324))3$E46:!43Z!%6%1%3D2@
M5$A)4Z!)4Z!!3J!/0DI%0U2@5TE42`TJH$-/3U)$24Y!5$53H$%4H"@K+RTQ
M+#`L,"PP*:`H,"PK+RTQ+#`L,"F@151#+@TJ#2J@;TZ@5$A%H$]42$52H$A!
M3D0LH%1(1:!/0U1!2$5$4D].H$E3H$5!4TE,6:!$15))5D5$#2J@1E)/3:!4
M2$6@0U5"12R@0EF@2D])3DE.1Z!42$6@0T5.5$524Z!/1J!%04-(H$9!0T4-
M*J!/1J!42$6@0U5"1:`H0U545$E.1Z!/1D:@5$A%H$-/4DY%4E,LH$E&H%E/
M5:!,24M%*2X-*J!T2%53H$E4H$E3H$5!4UF@14Y/54=(H%1/H$585$5.1*!4
M2$E3H$E.5$^@-&2@0EF@2D])3DE.1PTJH%1(1:!#14Y415)3H$]&H$5!0TB@
M,F2@1D%#1:!/1J!42$6@-&2@0U5"12R@5TA)0TB@3$5!1%,-*J!43Z!03TE.
M5%.@3$E+1:`H*R\M,2PK+RTQ+#`L,"F@*"LO+3$L,"PK+RTQ+#`IH$540RX-
M*@TJH'1(1:!&25)35*!-151(3T2@25.@04Y!3$]'3U53H%1/H$I/24Y)3D>@
M5$A%H$-%3E1%4J!/1@TJH$5!0TB@,V2@0U)/4U,M4T5#5$E/3J!/1J!42$6@
M-&2@0U5"12R@5TA%4D5!4Z!42$6@4T5#3TY$#2J@34542$]$H$E3+*!!4Z!7
M05.@4U1!5$5$+*!*3TE.24Y'H%1(1:!#14Y415)3H$]&H$5!0T@-*J`R9*!#
M4D]34RU314-424].H"TMH%1(1:!#14Y415)3H$]&H$-50D53H%9%4E-54Z!4
M2$6@0T5.5$524PTJH$]&H%!,04Y%4RX-*@TJH'=(24-(H$U%5$A/1*!43Z!5
M4T4_H*!T2$6@4T5#3TY$H$U%5$A/1*!'14Y%4D%415.@,C2@5D525$E#15,-
M*J!7251(H#DVH$Q)3D6@4T5'345.5%.@2D])3DE.1Z!42$5-H$%,3"Z@H&F@
M0T%.H%-%1:!72$%4H$E4#2J@3$]/2U.@3$E+1:!)3J!-6:!(14%$+*!IH$A!
M5D6@1%)!5TZ@0:!024-455)%+*!!3D2@:::!(059%#2J@5U))5%1%3J!$3U=.
MH$%,3*!42$6@4$])3E13H$%.1*!,24Y%H$-/3DY%0U1)3TY3+J"@84Q42$]5
M1T@-*J!)5*!214%,3%F@25.@4%)%5%19H%-44D%)1TA41D]25T%21*!43Z!6
M25-504Q)6D4LH$E4H%-%14U3#2J@3T)624]54Z!43Z!-1:!42$%4H$].H%1(
M1:!#3TU0551%4J!)5*!724Q,H$Q/3TN@3$E+1:!!H%)%04P-*J!-15-3+*!%
M4U!%0TE!3$Q9H$].H$I54U2@0:`Y-E@Y-J!'4DE$+*!%5D5.H%=)5$B@1$]4
M5$5$H$Q)3D53+@TJH&U/4D5/5D52+*!IH"I$3RJ@3TY,6:!(059%H#1+H%1/
MH%=/4DN@5TE42"R@04Y$H%1(1:`Q,BU3241%1`TJH$]"2D5#5*!724Q,H$)%
MH$&@1$]/6EF@5TE42*`Y-J!#3TY.14-424].4Z!!3D2@4U5#2"R@4T\-*J!I
MH$I54U2@55-%H%1(1:!/0DI%0U2@1E)/3:!42$6@1DE24U2@34542$]$+@TJ
M#2J@*'1(1:!314-/3D2@34542$]$H$E3H$%71E5,3%F@0T]/3*!43Z!42$E.
M2Z!!0D]55"R@5$A/54=(+`TJH$Y/5*!43Z!-14Y424].H$Y%050M3Z!+145.
MH%1/H$Q/3TN@050A*0TJ#61E9F]B,@TZ:6YI=`T@;&1A(",R,B`[.*`T9"R@
M-J`S9*`KH#B@3D]234%,4PT@8VUP('1O=&YU;0T@8F5Q(#IG<F]W(#MO3DQ9
MH%)53J!42$6@24Y)5*!23U5424Y%H$].0T4-#2!S=&$@=&]T;G5M#2!L9&$@
M(S@-('-T82!N=6UO8C$@.W!224U!4EF@*#1D*0T@;&1A(",Q-`T@<W1A(&YU
M;6]B,3(@.W-%0T].1$%26:`H,V0IH"N@3D]234%,4PT@;&1A(",R-"LQ,@T@
M<W1A(&YU;6-O;G,@.S,VH$-/3DY%0U1)3TY3#0T@;&1X(",V#2!L9&$@(R1F
M-B`[;4%+1:`S9*!/0DI%0U2@0:!,25143$6@4TU!3$Q%4@TZ;#$@<W1A(&]B
M;&ES='<K-RQX#2!D97@-(&)N92`Z;#$-#2J@<T54H%50H%1(1:!#3TY.14-4
M24].H$Q)4U13#2IO8C)C;VZ@+3Z@<&QI<W1W+*`S-J!#3TY.14-424].4Z`H
M,J!05%.@4$52H$-/3BD-*FYO<FTRH"T^H'!L:7-T>2R@H#0TH%!44Z!43U1!
M3`T@;&1Y(",W,2`[,S8J,@TZ;&]O<"!L9&$@;V(R8V]N+'D-('-T82!P;&ES
M='<L>0T@;&1A(&YO<FTR+'D-('-T82!P;&ES='DL>0T@9&5Y#2!B<&P@.FQO
M;W`-#2!J<W(@<V5T8V]N<PT-*@TJH'-%5*!72$E#2*!#3TY.14-424].4Z!7
M24Q,H%531:!$3U14142@3$E.15,-*J!I3J!42$E3H$-!4T4LH$Q)3D53H$-/
M3DY%0U1)3D>@0:!03TE.5*!43PTJH$%.3U1(15*@4$])3E2@5TE42*!.3TY:
M15)/H%<M0T]/4D1)3D%412X-*@T@;&1Y(",Q,2`[::!35%5#2Z!42$5-H$E.
MH%1(1:!&25)35`TZ;#(@;&1A(&1R87=C;VXL>2`[5%=%3%9%H%-,3U13#2!O
M<F$@(R0X,`T@<W1A(&1R87=C;VXL>0T@9&5Y#2!B<&P@.FPR#0TZ9W)O=PT@
M;&1A(",\=F5R=#(-('-T82!P;VEN=#$-(&QD82`C/G9E<G0R#2!S=&$@<&]I
M;G0Q*S$-(&QD>2`C,C(@.VY534)%4J!/1J!03TE.5%.@5$^@1U)/5Z`P+BYY
M+3$-(&QD82`C)&9F(#MG4D]7H$%,3*!#3T]21%.@15%504Q,60T@:G-R(&=R
M;W<-*@TJH'1(25.@3T)*14-4H$E3H$&@3$E45$Q%H%--04Q,15*@5$A!3J!4
M2$6@3U1(15)3+`TJH%-/H&F@0TA%052@0:!,25143$6@04Y$H$I535"@0D]4
M2*!/0DI%0U13H$]55`TJH$&@3$E45$Q%H$9!4E1(15*@3TY#1:!42$59)U)%
MH$1/3D6@1U)/5TE.1RX-*@TJH'-04D])3D<A#2H-(&QD82!O8FQI<W1W#2!C
M;7`@(S(T#2!B;F4@.F1O;F4-(&QD>"`C,3,-.FPS(&%S;"!O8FQI<W1W+'@-
M(&%S;"!O8FQI<W1X+'@@.T-,3U-%H%1/H%1(1:!%1$=%#2!A<VP@;V)L:7-T
M>2QX(#LH1$]73J!"6:!!H%))5D52/RD-(&%S;"!O8FQI<W1Z+'@-(&1E>`T@
M8G!L(#IL,PT-.F1O;F4@<G1S#0TJ#2J@;D585*!42$6@5D525$58H$Q)4U0-
M*@UV97)T,B!D9F(@)3`Q,#`P,#`P#2!D9F(@)3$Q,#`P,#`P#2!D9F(@)3`P
M,#$P,#`P#2!D9F(@)3`P,3$P,#`P#2!D9F(@)3`P,#`P,3`P#2!D9F(@)3`P
M,#`Q,3`P#2!D9F(@)3`P,#`P,#`Q#2!D9F(@)3`P,#`P,#$Q(#LT9*!615)4
M24-%4PT@9&9B("4P,#`Q,#`P,`T@9&9B("4P,#$Q,#`P,`T@9&9B("4P,#`P
M,#$P,`T@9&9B("4P,#`P,3$P,`T@9&9B("4P,#`P,#`P,0T@9&9B("4P,#`P
M,#`Q,2`[,V2@5D525$E#15.@*$Y/5$6@4T%-1:!!4Z!!0D]612D-(&1F8B`E

M,#`P,3`Q,#$-(&1F8B`E,#`P,3`Q,3$-(&1F8B`E,#`P,3$Q,#$-(&1F8B`E
M,#`P,3$Q,3$-(&1F8B`E,#`Q,3`Q,#$-(&1F8B`E,#`Q,3`Q,3$-(&1F8B`E
M,#`Q,3$Q,#$-(&1F8B`E,#`Q,3$Q,3$@.V%.1*!42$6@3D]234%,4Z`M/C(R
MH%1/5$%,#2H-*J!C3TY.14-424].4PTJ#6]B,F-O;B!H97@@,#`P,C`P,#,P
M,#`T,#`P-3`P,#8P,#`W(#LP,J`P,Z`P-*`P-:`P-J`P-PT@:&5X(#`Q,#(P
M,3`S,#$P-#`Q,#4P,3`V,#$P-R`[=$A%4T6@5TE,3*!"1:!$3U14140-(&AE
M>"`P,C`T,#(P-3`R,#8P,C`W#2!H97@@,#,P-#`S,#4P,S`V,#,P-PT@:&5X
M(#`T,#8P-#`W#2!H97@@,#4P-C`U,#<@.S1DH$-/3DY%0U1)3TY3H"@R-*!4
M3U1!3"D-#6]B,C)C;VX@:&5X(#`X,&$P.#!B,#@P8S`X,&0@.S@M,3"@."TQ
M,:`X+3$RH#@M,3,-(&AE>"`P.3!A,#DP8C`Y,&,P.3!D#2!H97@@,&$P8S!A
M,&0@.S$P+3$RH#$P+3$S#2!H97@@,&(P8S!B,&0@.S-DH$-/3E.@*#$RH%1/
M5$%,*0T-;F]R;3(@.VY/4DU!3%,K0T]24D534$].1$E.1Z!#3TY.14-424].
M4PT@9&9B(#(T(#MN54U"15*@3T:@-&2@0T].3D5#5$E/3E,-(&1F8B`T,B`[
M;E5-0D52H$]&H%!/24Y44Z!"14Q/5PT@:&5X(#`P,&4@.VY/4DU!3*!!5*!)
M3D1%6*`Q-*`]H#$L,2PQ#2!D9F(@,C0@.S@M,3`-(&1F8B`S,B`[,3`M,3(-
M(&1F8B`R-B`[,3(M.`T@:&5X(#`P,&8-(&1F8B`R-`T@9&9B(#,S#2!D9F(@
M,C<-(&AE>"`P,#$P#2!D9F(@,C4-(&1F8B`S-`T@9&9B(#(V#2!H97@@,#`Q
M,0T@9&9B(#(U#2!D9F(@,S4-(&1F8B`R-PT@:&5X(#`P,3(-(&1F8B`R.`T@
M9&9B(#,R#2!D9F(@,S`-(&AE>"`P,#$S#2!D9F(@,C@-(&1F8B`S,PT@9&9B
M(#,Q#2!H97@@,#`Q-`T@9&9B(#(Y#2!D9F(@,S0-(&1F8B`S,`T@:&5X(#`P
M,34-(&1F8B`R.0T@9&9B(#,U#2!D9F(@,S$-(&AE>"`P,#`P(#LT,J!03TE.
M5%.@24Z@3$E35`T-*BTM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2T-
M*@TJH&1E9F]B,0TJ#2J@<T54H%50H$]"2D5#5*`C,2R@0:!&3U525$5%3BU3
M241%1*!42$E.1PTJ#2J@=$A)4Z!)4Z!42$6@5TE,1*!/3D4ZH#,RH%!/24Y4
M4Z!!3D2@.3:@0T].3D5#5$E/3E.@24X-*J!42$6@-&2@3T)*14-4H$%,3TY%
M+J"@9T]21T5/55,N#2H-*J!T2$E3H$]"2D5#5*!)4Z!#3TY35%)50U1%1*!"
M6:!#3TY.14-424Y'H%1(1:!-24103TE.5%,-*J!/1J!%04-(H$Q)3D6@4T5'
M345.5*!/3J!42$6@0U5"12Z@H'1(25.@04U/54Y44Z!43PTJH%-%5%1)3D>@
M3TY%H$]&H%1(1:!#3T]21$E.051%4Z!/1J!!H$-50D6@5D525$58H%1/H%I%
M4D\L#2J@04YH]&H$-/55)31:!''259%4Z!&3U52H%!/24Y44Z!&3U*@14%#
M2*!&04-%H$]&H%1(10TJH$-50D4NH*!E6$%-4$Q%H#1DH$-/3U)$24Y!5$53
MH$%21:`H,2PQ+#$L,"F@*#$L+3$L,"PQ*0TJH"@M,2PP+"TQ+#IH540RX-
M*@TJH&E.H%1(4D5%H$1)345.4TE/3E,LH%1(25.@2$%3H%1(1:!%1D9%0U2@
M3T:@3$]04$E.1Z!42$4-*J!#3U).15)3H$]&H%1(1:!#54)%H$]&1J`H3T:@
M0T]54E-%H$Y/5*!!4Z!$4D%35$E#04Q,60TJH$%3H%=)5$B@5$A%H%1%5%)!
M1T].*2R@3$5!5DE.1Z!$24%-3TY$4Z!72$521:!42$6@0U5"10TJH$9!0T53
MH%53142@5$^@0D6@04Y$H%1224%.1TQ%4Z!72$521:!42$6@0T]23D524Z!5
M4T5$#2J@5$^@0D4N#2H-*J!T2$6@4T],242@3$E.15.@05)%H%1(1:`B24Y.
M15(BH$%.1*`B3U5415(BH#$T+4=/3E,L#2J@04Y$H%1(1:!$3U14142@3$E.
M15.@05)%H$%,3*!42$6@(DU)1$1,12*@,30M1T].4PTJH%1(052@64]5H$=%
M5*!)3BU"1517145.H%1(1:!)3DY%4J!!3D2@3U5415*@1U594RX-*@TJH&].
M1:!/5$A%4J!42$E.1Z!43Z!.3U1%H$E3H%1(052@5$A%H$Y/4DU!3%.@05)%
MH$&@3$E45$Q%#2J@1$E&1D5214Y4H$9/4J!42$E3H$].12Z@H'1(1:!(241$
M14Z@1D%#1:!!3$=/4DE42$V@1$5014Y$4PTJH$].H%1(1:!$3U2@4%)/1%5#
M5*!/1J!42$6@3D]234%,H%9%0U1/4J!7251(H$&@4$])3E0-*J!"14E.1Z!!
MH%!!4E1)0U5,05*@0T].4U1!3E0NH*!T2$%4H$E3+*!/3D6@3D]234%,H%9%
M0U1/4@TJH$E3H"@Q+#`L,"F@04Y$H$E4H$E3H$1/5%1%1*!)3E1/H"@Q+#$L
M,"DLH$=)5DE.1Z`Q+@TJH"@Q+#`L,"F@25.@0:!.3U)-04R@5D5#5$]2H$9/
M4J!42$6@15@M0U5"1:!&04-%H"A.3U<-*J!!H$1)04U/3D0I+J"@8:!.3U)-
M04R@5D5#5$]2H%1/H%1(1:!44DE!3D=,15,LH%1(3U5'2"P-*J!,3T]+4Z!,
M24M%H"@Q+#$L,2DLH$DN12Z@5TA%4D6@5$A%H$],1*!#54)%H%9%4E1%6*!5
M4T5$#2J@5$^@0D4NH*!W2$5.H$1/5%1%1*!)3E1/H"@Q+#$L,"F@5$A)4Z!'
M259%4Z`R+*!!H$1)1D9%4D5.5`TJH$-/3E-404Y4+J"@=$^@1DE8H%1(25.@
M55`LH%1(1:!.3U)-04Q3H$Y%142@5$^@0D6@2$%,5D5$+`TJH%1(052@25,L
MH$E.4U1%042@3T:@*#$L,2PQ*:!)5*!724Q,H$)%H"@Q+S(LH#$O,BR@,2\R
M*2X-*J!N3U>@5$A%H$1/5*!04D]$54-4H$E3H#&@04=!24XLH$%.1*!71:!$
M3TXG5*!.145$H%1/#2J@4U!%0TE!3"U#05-%H%1(1:!(241$14Z@1D%#1:!!
M3$=/4DE42$TN#2H-9&5F;V(Q#3II;FET#2!L9&$@(S4X(#LS,J`T9"R@,3*@
M,V2@*Z`Q-*!.3U)-04Q3#2!C;7`@=&]T;G5M#2!B97$@.F=R;W<@.V].3%F@
M4E5.H%1(1:!)3DE4H%)/551)3D6@3TY#10T-('-T82!T;W1N=6T-(&QD82`C
M,S(-('-T82!N=6UO8C$@.W!224U!4EF@*#1D*0T@;&1A(",Q,BLX*S8-('-T
M82!N=6UO8C$R(#MS14-/3D1!4EF@*#-D*:`KH$Y/4DU!3%,-(&QD82`C,3(P
M#2!S=&$@;G5M8V]N<R`[,3(PH$-/3DY%0U1)3TY3#0T@;&1X(",Q,@T@;&1A
M(",D9C@@.VU!2T6@,V2@3T)*14-4H$&@3$E45$Q%H%--04Q,15(-.FPQ('-T
M82!O8FQI<W1W*S,Q+'@-(&1E>`T@8FYE(#IL,0T-*J!S152@55"@5$A%H$-/
M3DY%0U1)3TZ@3$E35%,-*F]B,6-O;J`M/J!P;&ES='<LH#$R,*!#3TY.14-4
M24].4Z`H,J!05%.@4$52H$-/3BD-*FYO<FTQH"T^H'!L:7-T>2R@H#@PH%!4
M4Z!43U1!3`T@;&1Y(",P,`TZ;&]O<"!L9&$@;V(Q8V]N+'D-('-T82!P;&ES
M='<L>0T@;&1A(&YO<FTQ+'D-('-T82!P;&ES='DL>0T@:6YY#2!C<'D@(S(T
M,"`[,3(P*C(-(&)N92`Z;&]O<`T-(&IS<B!S971C;VYS#0TJ#2J@<T54H%=(
M24-(H$-/3DY%0U1)3TY3H%=)3$R@55-%H$1/5%1%1*!,24Y%4PTJH&E.H%1(
M25.@0T%312R@3$E.15.@0T].3D5#5$E.1Z!!H%!/24Y4H%1/#2J@04Y/5$A%
M4J!03TE.5*!7251(H$Y/3EI%4D^@5RU#3T]21$E.051%+@TJ#2!L9'D@(S(T
M(#MIH%-454-+H%1(14V@24Z@4TQ/5%.@,C0M-S$-.FPR(&QD82!D<F%W8V]N
M+'D-(&]R82`C)#@P#2!S=&$@9')A=V-O;BQY#2!I;GD-(&-P>2`C-S(-(&)N
M92`Z;#(-*@TJH'-%5*!54*!42$6@3D]234%,4PTJ#2J@=$A%4D6@05)%H%17
M3Z!31513H$]&H$Y/4DU!3%,NH*!T2$6@1DE24U2@4T54H$-/4E)%4U!/3D0-
M*J!43Z!42$6@5D525$E#15.@3T:@0:!#54)%H"A$259)1$5H)9H%173RF@
M04Y$H%1(10TJH%-%0T].1*!3152@3$E%H$].H%1(1:!#3T]21$E.051%H$%8
M15,N#2H-(&QD>2`C,C4V+3$R(#MS059%4Z!!H$)95$6@*%E/5R$A(2$A*0T@
M;&1X(",P-R`[9DE24U2@4T54#3IL-"!S='@@=&5M<#$@.VQ%5*!XH$))5#TP

MH$-/4E)%4U!/3D2@5$^@*S$O,@T@;'-R('1E;7`Q(#M!3D2@0DE4/3&@0T]2
M4D534$].1*!43Z!#3T]21*`M,2\R#2!L9&$@(S$R(#MC3T]21%.@34%8H$]5
M5*!!5*`R-`T@8F-C(#IC;VYT,2`[2$%,1J!/1J!72$E#2*!)4Z`Q,@T@='EA
M#3IC;VYT,2!S=&$@;V)L:7-T>BLT-"QX(#MN3U)-04Q3H%-405)4H$%4H%!/
M24Y4H#0T#2!L<W(@=&5M<#$-(&QD82`C,3(-(&)C8R`Z8V]N=#(-('1Y80TZ
M8V]N=#(@<W1A(&]B;&ES='DK-#0L>`T@;'-R('1E;7`Q#2!L9&$@(S$R#2!B
M8V,@.F-O;G0S#2!T>6$-.F-O;G0S('-T82!O8FQI<W1X*S0T+'@-(&1E>`T@
M8G!L(#IL-`T-(&QD82`C,C0@.VY/5Z!$3Z!314-/3D2@4T54#2!S=&$@;V)L
M:7-T>"LU,B`[*#$L,"PP*0T@<W1A(&]B;&ES='DK-3,@.R@P+#$L,"D-('-T
M82!O8FQI<W1Z*S4T(#LH,"PP+#$I#2!L9&$@(S(U-BTR-`T@<W1A(&]B;&ES
M='@K-34@.R@M,2PP+#`I#2!S=&$@;V)L:7-T>2LU-B`[*#`L+3$L,"D-('-T
M82!O8FQI<W1Z*S4W(#LH,"PP+"TQ*0T-.F=R;W<-(&QD82`C/'9E<G0Q#2!S
M=&$@<&]I;G0Q#2!L9&$@(SYV97)T,0T@<W1A('!O:6YT,2LQ#2!L9'D@(S0T
M(#MN54U"15*@3T:@4$])3E13H%1/H$=23U>@,"XN>2TQ#2!L9&$@(R1F9B`[
M9U)/5Z!!3$R@0T]/4D13H$5154%,3%D-(&IM<"!G<F]W#0TJ#2J@;D585*!4
M2$6@5D525$58H$Q)4U0-*@UV97)T,2!D9F(@)3`Q,#`P,3`Q(#LH,2PP+#$L
M,2D-(&1F8B`E,#$P,#`Q,3$@.R@Q+#`L,2PM,2D-(&1F8B`E,#$P,#$Q,3$@
M.R@Q+#`L+3$L+3$I#2!D9F(@)3`Q,#`Q,3`Q(#LH,2PP+"TQ+#$I#2!D9F(@
M)3`Q,#$P,#`Q(#LH,2PQ+#`L,2D-(&1F8B`E,#$P,3`P,3$@.T540PT@9&9B
M("4P,3$Q,#`Q,0T@9&9B("4P,3$Q,#`P,2`[-PT@9&9B("4P,3`Q,#$P,`T@
M9&9B("4P,3`Q,3$P,`T@9&9B("4P,3$Q,3$P,`T@9&9B("4P,3$Q,#$P,`T@
M9&9B("4Q,3`P,#$P,0T@9&9B("4Q,3`P,#$Q,0T@9&9B("4Q,3`P,3$Q,0T@
M9&9B("4Q,3`P,3$P,2`[,34-(&1F8B`E,3$P,3`P,#$-(&1F8B`E,3$P,3`P
M,3$-(&1F8B`E,3$Q,3`P,3$-(&1F8B`E,3$Q,3`P,#$-(&1F8B`E,3$P,3`Q
M,#`-(&1F8B`E,3$P,3$Q,#`-(&1F8B`E,3$Q,3$Q,#`-(&1F8B`E,3$Q,3`Q
M,#`@.S(S#2!D9F(@)3`P,#$P,3`Q#2!D9F(@)3`P,#$P,3$Q#2!D9F(@)3`P
M,#$Q,3$Q#2!D9F(@)3`P,#$Q,3`Q#2!D9F(@)3`P,3$P,3`Q#2!D9F(@)3`P
M,3$P,3$Q#2!D9F(@)3`P,3$Q,3$Q#2!D9F(@)3`P,3$Q,3`Q(#LS,2R@14Y$
MH$]&H#1DH%!/24Y44PT@9&9B("4P,#`P,#$P,0T@9&9B("4P,#`P,#$Q,0T@
M9&9B("4P,#`P,3$P,0T@9&9B("4P,#`P,3$Q,0T@9&9B("4P,#`Q,#`P,0T@
M9&9B("4P,#`Q,#`Q,0T@9&9B("4P,#`Q,#$P,`T@9&9B("4P,#`Q,3$P,"`[
M,SD-(&1F8B`E,#`Q,3`P,#$-(&1F8B`E,#`Q,3`P,3$-(&1F8B`E,#`Q,3`Q
M,#`-(&1F8B`E,#`Q,3$Q,#`@.S0S+*!%3D2@3T:@,V2@4$])3E13#3MN3U)-
M04Q3H%-%5*!54*!"6:!23U5424Y%H$%"3U9%#0TJ#2J@8T].3D5#5$E/3E,-
M*@UO8C%C;VX@:&5X(#`P,#@P,#!B,#`P-#`P,#<@.S`M.*`P+3$QH#`M-*`P
M+3<-(&AE>"`P,3`X,#$P8C`Q,#4P,3`V(#LQ+3B@,2TQ,:`Q+36@,2TV#2!H
M97@@,#(P.3`R,&$P,C`U,#(P-@T@:&5X(#`S,#0P,S`W,#,P.3`S,&$-(&AE
M>"`P-#`X,#0P.0T@:&5X(#`U,#@P-3`Y#2!H97@@,#8P83`V,&(-(&AE>"`P
M-S!A,#<P8B`[;U5415*@,3(M1T].+*`R-*!#3TY3#2!H97@@,#`Q.#`P,6,@
M.S`M,C2@,"TR.`T@:&5X(#`Q,3DP,3%D(#LQ+3(UH#$M,CD-(&AE>"`P,C%A
M,#(Q90T@:&5X(#`S,6(P,S%F#2!H97@@,#0Q.#`T,6(@.V%,3*!42$531:!-
M241$3$6@1U594Z!!4D4-(&AE>"`P-3$Y,#4Q82`[1$]45$5H-/3DY%0U1)
M3TY3#2!H97@@,#8Q9#`V,64-(&AE>"`P-S%C,#<Q9@T@:&5X(#`X,3@P.#$Y
M(#LX+3(TH#@M,C4-(&AE>"`P.3%A,#DQ8@T@:&5X(#!A,64P83%F#2!H97@@
M,&(Q8S!B,60@.S$Q+3(XH#$Q+3(Y+*!O551%4BT^34E1Q%+*`R-*!#3TY3
M#2!H97@@,&,Q.#!C,6,-(&AE>"`P9#$Y,&0Q9`T@:&5X(#!E,6$P93%E(#LQ
M-"TR-J`Q-"TS,`T@:&5X(#!F,6(P9C%F#2!H97@@,3`Q.#$P,6(-(&AE>"`Q
M,3$Y,3$Q80T@:&5X(#$R,60Q,C%E#2!H97@@,3,Q8S$S,68-(&AE>"`Q-#$X
M,30Q.0T@:&5X(#$U,6$Q-3%B#2!H97@@,38Q93$V,68-(&AE>"`Q-S%C,3<Q
M9"`[,C,M,CB@,C,M,CDLH$U)1$1,12T^24Y.15(LH#(TH$U/4D6@0T].4PT@
M:&5X(#!C,30P8S$W,&,Q,#!C,3,@.V%.1*!&24Y!3$Q9+*!42$6@24Y.15*@
M0T].4PT@:&5X(#!D,30P9#$W,&0Q,3!D,3(-(&AE>"`P93$U,&4Q-C!E,3$P
M93$R#2!H97@@,&8Q,#!F,3,P9C$U,&8Q-@T@:&5X(#$P,30Q,#$U#2!H97@@
M,3$Q-#$Q,34-(&AE>"`Q,C$V,3(Q-PT@:&5X(#$S,38Q,S$W(#MI3DY%4BR@
M5$A%H$Q!4U2@,C2@0T].3D5#5$E/3E,-(#MF3U*@0:!43U1!3*!/1J`Y-J`T
M9*!#3TY.14-424].4PUO8C$R8V]N(&AE>"`R,#(T,C`R."`[,S(M,S:@,S(M
M-#"@*#-DH$-/3DY%0U1)3TY3*0T@:&5X(#(P,C8R,#)A(#LS,BTS.*`S,BTT
M,J`H3$%35*!"552@3D]4H$Q%05-4*0T@:&5X(#(Q,C4R,3(Y(#LS,RTS-Z`S
M,RTT,0T@:&5X(#(Q,C8R,3)A#2!H97@@,C(R-#(R,C@-(&AE>"`R,C(W,C(R
M8@T@:&5X(#(S,C4R,S(Y#2!H97@@,C,R-S(S,F(-(&AE>"`R-#(V,C0R-PT@
M:&5X(#(U,C8R-3(W#2!H97@@,C@R83(X,F(-(&AE>"`R.3)A,CDR8B`[,C2@
M34]21:`S9*!#3TY.14-424].4RR@1D]2H$&@5$]404P-(#M/1J`Q,C"@0T].
M3D5#5$E/3E,LH$I54U2@4TA9H$]&H%1(10T@.V)M::!-05)+(0T-;F]R;3$@
M.VY/4DU!3%,K0T]24D534$].1$E.1Z!#3TY.14-424].4PT@9&9B(#DV(#MN
M54U"15*@3T:@-&2@0T].3D5#5$E/3E,-(&1F8B`W."`[;E5-0D52H$]&H%!/
M24Y44Z!"14Q/5PT@9&9B(#`P#2!D9F(@-#0@.VY/4DU!3*!!5*!)3D1%6*`T
M-*`]H#$O,BPQ+S(L,2\R#2!D9F(@.38@.S,R+3,V#2!D9F(@,3$R(#LS-BTS
M.`T@9&9B(#DX(#LS."TS,@T@:&5X(#`P,F0@.VE.1$58H#0U#2!D9F(@,3`P
M#2!D9F(@,3$T#2!D9F(@,3`R#2!H97@@,#`R92`[-#8-(&1F8B`Q,#0-(&1F
M8B`Q,3,-(&1F8B`Q,#8-(&AE>"`P,#)F(#LT-PT@9&9B(#$P.`T@9&9B(#$Q
M-0T@9&9B(#$Q,`T@:&5X(#`P,S`@.S0X#2!D9F(@.3<-(&1F8B`Q,38-(&1F
M8B`Y.0T@:&5X(#`P,S$@.S0Y#2!D9F(@,3`Q#2!D9F(@,3$X#2!D9F(@,3`S
M#2!H97@@,#`S,B`[-3`-(&1F8B`Q,#4-(&1F8B`Q,3<-(&1F8B`Q,#<-(&AE
M>"`P,#,S(#LU,0T@9&9B(#$P.0T@9&9B(#$Q.0T@9&9B(#$Q,0T@:&5X(#`P
M,S0@.S4R+*!42$6@1$E!34].1%,-(&1F8B`Q,3(-(&1F8B`Q,30-(&1F8B`Q
M,34-(&1F8B`Q,3,-(&AE>"`P,#,U(#LU,PT@9&9B(#DX#2!D9F(@,3`R#2!D
M9F(@,3`S#2!D9F(@.3D-(&AE>"`P,#,V(#LU-`T@9&9B(#DV#2!D9F(@,3`T
M#2!D9F(@,3`U#2!D9F(@.3<-(&AE>"`P,#,W(#LU-0T@9&9B(#$Q-@T@9&9B
M(#$Q.`T@9&9B(#$Q.0T@9&9B(#$Q-PT@:&5X(#`P,S@@.S4V#2!D9F(@,3`V
M#2!D9F(@,3$P#2!D9F(@,3$Q#2!D9F(@,3`W#2!H97@@,#`S.2`[-3<LH$Q!
M4U2@3TY%(0T@9&9B(#$P,`T@9&9B(#$P.`T@9&9B(#$P.0T@9&9B(#$P,0T@

M:&5X(#`P,#`@.S8J-BLX*C4K,CTW.*!03TE.5%.@24Z@3$E35`WE79*LFS[I
MDJ:IJG.:Y*GN;IY5V2K.M:TI*FN:IK5V2K.M:TI*FN:YI5V2K.M:TI*FN:IS
M5V2K)L^Z9*FJ:ISG*2I[F^:E=DJSK6M*2IKFJ;U=DJSK6M*2IKFN=U=DJSK6
MM+-5*4H`<"HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!-
M55-)0RY3#2H-*J!O4BR@5$^@0D6@34]21:!04D5#25-%+*!P4D5,541%H",R
MH$923TV@=$A%H'=%3$PM#2J@=$5-4$52142@:TQ!5DE%4BR@0EF@:BYS+J!B
M04-(+@TJ#2J@=$A)4Z!)4Z!42$6@34%)3J!-55-)0Z!23U5424Y%H$9/4J!4
M2$6@-$N@1$5-3RX-*J!T2$6@355324.@25.@1%))5D5.H$)9H$%.H$E.5$52
M4E505*!'14Y%4D%4142@0ED-*J!V:6,NH*!AH%-)35!,1:!#3U5.5$1/5TZ@
M5$E-15*@25.@1$5#4D5-14Y4140-*J!53E1)3*!:15)/H$E3H%)%04-(140L
MH$%4H%=(24-(H%!/24Y4H%1(1:!.15A4#2J@3D]41:!)4Z!03$%9140LH$%.
M1*!)1J!.14-%4U-!4EF@5$A%H$Y%6%2@3D]410TJH"A/4J!.3U1%4RF@05)%
MH%)%042@24XNH*!F3U*@34]21:!$151!24Q3+*!3146@5$A%#2J@0T]12-
M*@TJH'-415!(14Z@;"Z@:E5$1*"@-B\R-R\Y-J"@/"TM+:`H9$]73J!43Z!4
M2$6@5TE212R@14@_*0TJ#2!D<VL@(DU54TE#(@T-(')E;`T-*@TJH'9!4DE!
M0DQ%4PTJ#6-O=6YT97(@97%U("0P,R`[9%52051)3TZ@0T]53E1$3U=.#6EN
M9&5X(&5Q=2`D,#0@.VE.1$58H$E.5$^@3D]41:!404),1:`H3D]4H9215I
M#79O:6-E<R!E<74@)#`U(#MT14Q,4Z!72$E#2*!63TE#15.@05)%H$%#5$E6
M10UN=6UN;W1E<R!E<74@)#`V(#MN54U"15*@3T:@3D]415.@4D5!1*!)3J!!
M5*!!H%1)344-;F]T97!T<B!E<74@)#`W(#LR+4)95$6@4$])3E1%4J!43Z!-
M55-)0Z!$051!#61U<F%T:6]N(&5Q=2`D,#D@.V154D%424].H$]&H$5!0TB@
M3D]410UV;VQU;64@97%U("0P82`[9%5(#6=A=&4Q(&5Q=2`D,&(@.W9!3%5%
MH%1/H$=!5$4O54Y'051%H%9/24-%H#&@5TE42`UG871E,B!E<74@)#!C(#MS
M04U%H$9/4J!63TE#1:`R#6=A=&4S(&5Q=2`D,&0@.W-!346@1D]2H%9/24-%
MH#,-#2H-*J!S3TU%H$%715-/344M4$]34U5-H$-/3E-404Y44PTJ#79O:6-E
M,2!E<74@)&,R,#`@.V-54E)%3E2@3D]415.@1D]2H%9/24-%H#$-=F]I8V4R
M(&5Q=2`D8S(Q,"`[8U524D5.5*!.3U1%4Z!&3U*@5D])0T6@,@UV;VEC93,@
M97%U("1C,C(P(#MH34U-32XN+@UF<F5Q;&\@97%U("1C,S`P(#MF4D51545.
M0UF@5$%"3$4-9G)E<6AI(&5Q=2`D8S,X,`T-<F5S="!E<74@,#`@.W1214%4
MH$Y/5$6@6D523Z!!4Z!!H%)%4U0-(#LH5TA)0TB@2E535*!53BU'051%4Z!4
M2$6@5D])0T4I#0UR97-T87)T(&5Q=2`D.#`@.W-014-)04R@355324,M1$%4
M0:!43TM%3@US971D=7(@97%U("0X,2`[<T54+4154D%424].H%1/2T5.#7-E
M='9O;"!E<74@)#@R(#MS150M5D],54U%H%1/2T5.#7-E=&YU;2!E<74@)#@S
M(#MS152@3E5-0D52H$]&H$Y/5$53H%1/2T5.#7-E='8Q(&5Q=2`D.#0@.W9/
M24-%+3&@3TY,6:!43TM%3@US971V,3(@97%U("0X-2`[=D])0T6@,:!!3D2@
M,@US971V,3(S(&5Q=2`D.#8-#65I9VAT:"!E<74@-R`[9%52051)3TZ@3T:@
M0:!325A4145.5$B@3D]41:`Z*0UP<F5S=&\@97%U(#4@.S$V5$B@3D]41:!)
M3J!P4D535$^@4T5#5$E/3@UA9&%G:6\@97%U(#$X(#LQ-E1(H$Y/5$6@24Z@
M841!1TE/H%-%0U1)3TX-9B!E<74@,3(@.V9/4E1%#7-F(&5Q=2`Q-"`[(G-&
M3U)46D%.1$\B#69F(&5Q=2`Q-2`[9$]50DQ%+49/4E1%#7`@97%U(#@@.W!)
M04Y/#0TJ#2J@<T]-1:!(04Y$62U$04Y$6:!C-C0M4D5,051%1*!#3TY35$%.
M5%,-*@T-<VED(&5Q=2`D9#0P,`UV,2!E<74@<VED#78R(&5Q=2!S:60K-PUV
M,R!E<74@=C(K-PT-*J!K15).04P-#7-C;FME>2!E<74@)&5A.#<-9V5T:6X@
M97%U("1F,3-E#6-H<F]U="!E<74@)&9F9#(-#2HM+2TM+2TM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM#2H]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]
M/3T]#2J@;D]4H$%34T5-0DQ%1`T@9&\@,"`[9$].)U2@05-314U"3$4-('-E
M:0T@:G-R(&UU<VEN:70-(&-L:0TZ=&5S="!J<W(@<V-N:V5Y(#MJ55-4H%1%
M4U2@24:@5T6@05)%H$-!3$Q)3D<-(&IS<B!G971I;B`[24Y415)255!4H$-/
M4E)%0U1,62R@151#+@T@8VUP(",P,`T@8F5Q(#IT97-T#2!J<W(@8VAR;W5T
M#2!J;7`@.G1E<W0-(&9I;@TJ/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]
M/3T]/0T-*BTM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2T-*@TJH&E.
M251)04Q)6D6@24Y415)255!4H%)/551)3D4-*@TJH&Y/5$6@5$A!5*!#04Q,
M24Y'H%)/551)3D6@35535*!S96DO8VQIH"$A(0TJ#0UM=7-I;FET(&5N="`[
M<T54H%50H$E.5$524E505`T-(&QD>"`C,3$@.W-%5*!54*!.3U1%H$9215%5
M14Y#6:!404),15,-(#MS24Y#1:!IH%-#4D57142@55"@04YHU!1$4-(#M%
M5D52651(24Y'H$%.H$]#5$%61:!43T^@2$E'2"P-(#MIH$-/35!%3E-!5$6@
M2$521:!)3J!42$6@5$%"3$4-(#M315154*!624&@04Z@;'-R+W)O<@T@.V]2
M+*!)1J!93U6@3$E+12R@252@4T]53D13H$)%5%1%4@T@.T%.H$]#5$%61:!,
M3U=%4J!7251(H%1(25.@24Y35%)5345.5`TZ;#$@;&1A(&AI=&%B+'@-(&QS
M<@T@<W1A(&9R97%H:2LX-"QX(#MS5$]21:!!5*!43U"@3T:@5$%"3$4-(&QD
M82!L;W1A8BQX#2!R;W(-('-T82!F<F5Q;&\K.#0L>`T@9&5X#2!B<&P@.FPQ
M#2!L9'@@(S@S(#MN3U>@15A414Y$H%1(1:!404),10TZ;#(@;&1A(&9R97%H
M:2LQ,BQX(#M$3U=.5T%21%,LH$1)5DE$24Y'H$)9#2!L<W(@.U173Z!43Z!'
M152@3T-4059%4PT@<W1A(&9R97%H:2QX#2!L9&$@9G)E<6QO*S$R+'@-(')O
M<@T@<W1A(&9R97%L;RQX#2!D97@-(&)P;"`Z;#(@.W1(052@3U5'2%2@5$^@
M1$^@250A#0UN;W1E:6YI=`T@;&1X(",R,"`[:4Y)5$E!3$E:1:!S:60-.FQO
M;W`@;&1A('-I9&EN:70L>`T@<W1A('-I9"QX#2!D97@-(&)P;"`Z;&]O<`T-
M(&QD82`C,38-('-T82!N=6UN;W1E<R`[<D5!1*!)3J`Q-J!.3U1%4Z!!5*!!
MH%1)344-(#M)3DE424%,3%D-('-T82!I;F1E>`T@;&1A(",P,`T@<W1A('9O
M;'5M90T@;&1A('-I9&EN:70K-`T@<W1A(&=A=&4Q#2!L9&$@<VED:6YI="LQ
M,0T@<W1A(&=A=&4R#2!L9&$@<VED:6YI="LQ.`T@<W1A(&=A=&4S#0TJ/3T]
M/3T]/3T]/3T]/3T]/3T]/3T]/3T]/3T]/0T@9&\@,"`[9$].)U2@05-3
M14U"3$4-(&QD82`C,34-('-T82!S:60K,C0-(&QD82!G871E,0T@<W1A('8Q
M*S0-(&QD>"`C,#`@.W1%4U2@5$A%H$Y/5$6@5$%"3$4-('-T>"!C;W5N=&5R
M#3IL,R!L9&$@9G)E<6QO+'@-('-T82!V,0T@;&1A(&9R97%H:2QX#2!S=&$@
M=C$K,0T@;&1Y(",Q,`TZ;#0@9&5C(&-O=6YT97(@.T1%3$%9H$Q/3U`-(&)N
M92`Z;#0-(&1E>0T@8FYE(#IL-`T@:6YX#2!C<'@@(SDV#2!B;F4@.FPS#2!L
M9&$@9V%T93$-(&%N9"`C)&9E#2!S=&$@=C$K-`T@9FEN#2H]/3T]/3T]/3T]
M/3T]/3T]/3T]/3T]/3T]/3T]/3T]#0T@;&1A(",\;F]T961A=&$-('-T82!N
M;W1E<'1R(#MP3TE.5$52H$E.5$^@3D]41:!$051!#2!L9&$@(SYN;W1E9&%T
M80T@<W1A(&YO=&5P='(K,0T-(&QD82`C/&UU<W!L87D-('-T82`D,#,Q-"`[

```
M<T54H$Q/0T%424].H$]&H$E.5$524E505`T@;&1A(",^;75S<&QA>0T@<W1A
M("0P,S$U#0T@;&1A(",D,#`-('-T82`D9&,P92`[:TE,3*!C:6&@24Y415)2
M55!4#0T@;&1A(",D,#$-('-T82!C;W5N=&5R#2!S=&$@9'5R871I;VX-('-T
M82`D9#`Q82`[94Y!0DQ%H%)!4U1%4J!)3E1%4E)54%0-('-T82`D9#`Q.2`[
M<D%35$52H$-/35!!4D4-('-T82`D9#`Q,B`[<T54H$Q)3D6@5$^@24Y415)2
M55!4H$%4#2`[*$Q)3D6@,:!/4J!,24Y%H#(U-Z!)4Z!/2RD-*J!L9&&@)&0P
M,3$-*J!A;F2@(R0W9J`[>D%0H%504$52H'9I8Z!"250-*J!S=&&@)&0P,3$-
M(')T<PT-*BTM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2T-*@TJH'-/
M346@24Y)5$E!3$E:051)3TZ@1$%40:!&3U*@<VEDH$%.1*!.3U1%H$9215%5
M14Y#60TJH%1!0DQ%+@TJ#2J@:E535*!!H$)/4DE.1Z!/3*!305=43T]42"X-
M*@US:61I;FET(&AE>"`P,#`P,#`P,#(Q,CAA."`[=D])0T6@,:!,3Z!(2:!0
M3$^@4$A)H$=!5$6@04134@T@:&5X(#`P,#`P,#`P,C$R86%D(#MV3TE#1:`R
M#2!H97@@,#`P,#`P,#`R,#)A860@.W9/24-%H#,-#2J@=$A%4T6@05)%H$92
M15%514Y#2453H$]55*!/1J!P<F>@1D]2H$5154%,H%1%35!%4D5$#2J@4T-!
M3$4NH*!E455!3*!414U015)%1*!)4Z!!H$1204>@0E54H$E4H$]51TA4H%1/
MH%=/4DN@;VLN#0UL;W1A8B!D9F(@,S`L,C0L,3,Y+#$R-BPR-3`L-BPQ-S(L
M,C0S+#(S,"PQ-#,L,C0X+#0V#0UH:71A8B!D9F(@,3,T+#$T,BPQ-3`L,34Y
M+#$V."PQ-SDL,3@Y+#(P,"PR,3(L,C(U+#(S."PR-3,-#2HM+2TM+2TM+2TM
M+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!T2$6@34%)3J!-55-)0Z!03$%9
M15*@4D]55$E.12X-*@T-*@TJH&9)4E-4+*!!H$9%5Z!.249462U34$E&1E19
MH$U!0U)/4PTJ#0TJH'!L87EN;W1EH$1/15.@5TA!5*!)5*!305E3+J"@=$A2
M146@5D%224%"3$53H$%21:!005-3140-*J!)3CJ@5$A%H$9)4E-4H$E3H'9O
M:6-E,:!V;VEC93*@3U*@=F]I8V4S+*!!3D2@25,-*J!42$6@3D585*!.3U1%
MH%1/H$)%H%!,05E%1"Z@H'1(1:!314-/3D2@25.@=C&@=C*@=C,-*J!72$E#
M2*!!4D6@5$A%H$%04%)/4%))051%H'-I9*!,3T-!5$E/3E.@5$^@4U1/4D6@
M4U151D8N#2J@=$A%H%1(25)$H$E3H&=A=&4Q+*!%5$,N+*!72$E#2*!%4U-%
M3E1)04Q,6:!414Q,4Z!72$542$52#2J@5$^@1T%41:!/4J!53D=!5$6@0:!.
M3U1%+@T-<&QA>6YO=&4@;6%C#2!L9&$@73$L>"`[=F]I8V4SH$540RZ@0T].
M5$%)3E.@24Y$15B@24Y43PT@=&%Y(#M.3U1%H$9215%514Y#6:!404),10T@
M;&1A(&9R97%L;RQY#2!S=&$@73(@.W-43U)%H$E4H$E.H'-I9`T@;&1A(&9R
M97%H:2QY(#ML3R]H2:!&4D51545.0UD-('-T82!=,BLQ#2!L9&$@73,@.V=!
M5$4O54Y'051%H$Y/5$4-('-T82!=,BLT#2`\/#P-#2J@:52@25.@4TE-4$Q%
M4J!43Z!0552@5$A%H$5.1*!!5*!42$6@0D5'24Y.24Y'H#HI#0UA;&QD;VYE
M('-T>"!I;F1E>`T@<&QA#2!T87D@.W)%4U1/4D6@4D5'25-415)3#2!P;&$-
M('1A>`T@<&QA#2!R=&D-*@TJH&Y/5Z!42$6@34%)3J!03$%915*@4D]55$E.
M10TJ#6UU<W!L87D-(&QD82`D9#`Q.0T@<W1A("1D,#$Y(#MC3$5!4J!)3E1%
M4E)54%2@1DQ!1PT@;&1X(&EN9&5X(#MI3D1%6*!)3E1/H$Y/5$6@5$%"3$4-
M(&1E8R!C;W5N=&5R#2!B;F4@86QL9&]N90T@;&1A(&1U<F%T:6]N(#MR15-%
M5*!#3U5.5$52#2!S=&$@8V]U;G1E<@T@8FET('9O:6-E<R`[=TA)0TB@5D])
M0T53H$%21:`B04-4259%(C\-(&)P;"!P;&%Y,B`[0DE4-ST^5D])0T4SH$))
M5#8]/E9/24-%,@T-<&QA>3,@/CX^('!L87EN;W1E+'9O:6-E,SMV,SMG871E
M,PUP;&%Y,B!B=F,@<&QA>3$-(#X^/B!P;&%Y;F]T92QV;VEC93([=C([9V%T
M93(-<&QA>3$@/CX^('!L87EN;W1E+'9O:6-E,3MV,3MG871E,0T@;&1A('9O
M;'5M90T@<W1A('-I9"LR-`T-(&EN>`T@8W!X(&YU;6YO=&5S(#ME251(15*@
M,3:@3U*@,0T@8FUI(&%L;&1O;F4-#2J@;4%)3J!005)315*@4D]55$E.12Z@
MH&1!5$&@25.@14E42$52H$%.H$E.1$58H$E.5$\-*J!42$6@3D]41:!&4D51
M545.0UF@5$%"3$6@*%1(55,LH$&@3D]412F@3U*@14Q31:!!#2J@4U!%0TE!
M3*!#2$%204-415(NH*!S4$5#24%,H$-(05)!0U1%4E.@2$%61:!42$4-*J!(
M24=(H$))5*!3152@04Y$H$A!5D6@5$A%H$9/3$Q/5TE.1Z!-14%.24Y'4SH-
M*J"@H*`MH')%4U1!4E2@355324,-*J"@H*`MH&5615)9H$Y/5$6@049415*@
M5$A)4Z!(05.@1%52051)3TZ@6%@-*J"@H*`MH'-%5*!63TQ5346@5$^@6%@-
M*J"@H*`MH'9/24-%H#&@3TY,6:`H;D^@1$%40:!&3U*@5D])0T6@,J!/4J`S
M*0TJH*"@H"V@=D])0T4QH$%.1*`RH$].3%D-*J"@H*`MH'9/24-%H#&@,J!!
M3D2@,PTJH*"@H"V@;D]415.@05)%H%)%042@24Z@3TY%H$%4H$&@5$E-1:!&
M4D]-H$Y/5Z!/3@TJ#2J@9DE24U2@04Q,H$193D%-24-3H$4N1RZ@4U!%0TE!
M3*!#2$%204-415)3H$%21:!214%$H$E.+`TJH%1(14Z@5$A%H$Y%6%2@3D]4
M15.@05)%H%)%042@24XNH*!T2$531:!604Q515.@5TE,3*!"10TJH%501$%4
M142@24Y43Z!S:62@5$A%H&YE>'2@5$E-1:!!4D]53D0N#0UP87)S90T@;&1Y
M(",P,`TZ;&]O<"!L9&$@*&YO=&5P='(I+'D-(&)P;"!R96%D;F]T92`[:$E'
M2*!"252@0TQ%05*@345!3E.@3D]41:!$051!#0T@8VUP("-R97-T87)T#2!B
M;F4@.F1U<@T@:G-R(&YO=&5I;FET(#MR14E.251)04Q)6D4LH%-+25!024Y'
MH%%1!0DQ%4PT@;&1X(",P,`T@:FUP(&%L;&1O;F4-.F1U<B!C;7`@(W-E=&1U
M<B`[8TA!3D=%H$154D%424].#2!B;F4@.G9O;`T@:6YY#2!L9&$@*&YO=&5P
M='(I+'D@.V=%5*!$55)!5$E/3E.@3T:@1D],3$]724Y'H$Y/5$53#2!S=&$@
M9'5R871I;VX-(&)N92`Z9&]N90TZ=F]L(&-M<"`C<V5T=F]L(#MS152@3D57
MH%9/3%5-10T@8FYE(#IV,0T@:6YY(#MG152@5D],54U%H%1/H%-%5*!43PT@
M;&1A("AN;W1E<'1R*2QY#2!S=&$@=F]L=6UE#2!B;F4@.F1O;F4-.G8Q(&-M
M<"`C<V5T=C$@.W9/24-%H#&@3TY,60T@8FYE(#IV,@T@;&1A(",P,`T@<W1A
M('9O:6-E<PT@8F5Q(#ID;VYE(#LH4D5!3$Q9H$1/3B=4H$Y%142@5$A%4T6@
M0E)!3D-(15,I#3IV,B!C;7`@(W-E='8Q,@T@8FYE(#IV,PT@;&1A(",D-#`@
M.W-%5*!"252@4TE8#2!S=&$@=F]I8V5S#2!B;F4@.F1O;F4-.G8S(&-M<"`C
M<V5T=C$R,PT@8FYE(#IN97=N=6T-(&QD82`C)&,P(#MB252@4TE8H$%.1*!3
M159%3@T@<W1A('9O:6-E<PT@8FYE(#ID;VYE#3IN97=N=6T@;&1A(",P,2`[
M:4:@5T6@1T]4H$A%4D6@5$A%3J!42$E3H$E3#2!S=&$@;G5M;F]T97,@.U1(
M1:!/3DQ9H$].1:!,1494(0TZ9&]N92!I;GD@.W1(052@5T%3H%-/H$U50TB@
M1E5.+BXN#2!B;F4@.FQO;W`@.TQ%5"=3H$1/H$E4H$%'04E.+@T-<F5A9&YO
M=&4@.VY/5Z!214%$H$E.H$Y/5$6@1$%400T@;&1X(&YU;6YO=&5S#2!C<'@@
M(S$V#2!B97$@<F5A9#$V(#MR3U5424Y%H%1/H%)%042@,3:@3D]415.@052@
M0:!424U%#2`[;U1(15)725-%+*!214%$H$].1:!.3U1%H$%4H$&@5$E-10UR
M96%D,2!C;7`@(W)E<W0-(&)N92`Z=C$-(&QD82!G871E,2`[<F5S=*!724Q,
MH$U%04Z@54Y'051%H$Y/5$4-(&%N9"`C)&9E#2!S=&$@9V%T93$-(&)N92`Z
```

M=C(-.G8Q('-T82!V;VEC93$-(&QD82!G871E,0T@;W)A(",D,#$-('-T82!G
M871E,0TZ=C(@8FET('9O:6-E<R`[84Y9H%9/24-%H#*@1$%40:!43Z!214%$
MH$E./PT@8G9C(&YO=&5D;VYE(#MN3Z!63TE#1:`RH$U%04Y3H$Y/H%9/24-%
MH#,-(&EN>0T@;&1A("AN;W1E<'1R*2QY(#MO5$A%4E=)4T4LH%)%042@24Z@
M3D]410T@8VUP("-R97-T#2!B;F4@.FYO<&4R#2!L9&$@9V%T93(-(&%N9"`C
M)&9E#2!S=&$@9V%T93(-(&)N92`Z=C,-.FYO<&4R#2!S=&$@=F]I8V4R#2!L
M9&$@9V%T93(-(&]R82`C)#`Q#2!S=&$@9V%T93(-.G8S(&)I="!V;VEC97,@
M.W)%042@24Z@5D])0T6@,Z!$051!/PT@8G!L(&YO=&5D;VYE#2!I;GD-(&QD
M82`H;F]T97!T<BDL>0T@8VUP("-R97-T#2!B;F4@.FYO<&4S#2!L9&$@9V%T
M93,-(&%N9"`C)&9E#2!S=&$@9V%T93,-(&)N92!N;W1E9&]N90TZ;F]P93,-
M('-T82!V;VEC93,-(&QD82!G871E,PT@;W)A(",D,#$-('-T82!G871E,PT-
M;F]T9610;F4@:6YY("`[<$])3E2@5$^@3D585*!0245#1:!/1J!$051!#2!T
M>6$@.T%.1*!54$1!5$6@3D]41:!03TE.5$52#2!C;&,-(&%D8R!N;W1E<'1R
M#2!S=&$@;F]T97!T<@T@8F-C(#IC;VYT#2!I;F,@;F]T97!T<BLQ#3IC;VYT
M(&QD>"`C,#`@.W)%4T54H$E.1$58#2!J;7`@86QL9&]N92`[84YH=%5*!4
M2$6@2$5#2Z!/552@3T:@2$5212$-*@TJH&U50TB@3T:@5$A%H%!)14-%H$-/
M3E-)4U13H$]&H$)23TM%3J`T+4Y/5$6@0TA/4D13+`TJH$)23TM%3J!!0U)/
M4U.@1D]54J!31513H$]&H$9/55*@3D]415.@14%#2"Z@H'1(55,-*J!325A4
M145.H$Y/5$53H$-!3J!"1:!34$5#249)142@052@0:!424U%H%=)5$@-*J!*
M55-4H$9/55*@1$%40:!604Q515,LH$%.1*!42$E3H%)/551)3D6@2$%.1$Q%
M4PTJH%1(25.@0T%312X-*@UR96%D,38-('-T82!V;VEC93$@.V9)4E-4H$Y/
M5$6@3T:@0TA/4D2@1T]%4Z!43PT@.TY/5$53H#&@04Y$H#D-(&EN>0T@;&1A
M("AN;W1E<'1R*2QY#2!S=&$@=F]I8V4Q*S$-('-T82!V;VEC93$K,PT@<W1A
M('9O:6-E,2LU#2!S=&$@=F]I8V4Q*S<-(&EN>0T@;&1A("AN;W1E<'1R*2QY
M#2!S=&$@=F]I8V4Q*S(-('-T82!V;VEC93$K-@T@:6YY#2!L9&$@*&YO=&5P
M='(I+'D-('-T82!V;VEC93$K-`T-(&EN>2`[;D]7H%)%042@24Z@5D])0T6@
M,J!'55E3#2!L9&$@*&YO=&5P='(I+'D-('-T82!V;VEC93(-(&EN>0T@;&1A
M("AN;W1E<'1R*2QY#2!S=&$@=F]I8V4R*S$-('-T82!V;VEC93(K,PT@<W1A
M('9O:6-E,BLU#2!S=&$@=F]I8V4R*S<-(&EN>0T@;&1A("AN;W1E<'1R*2QY
M#2!S=&$@=F]I8V4R*S(-('-T82!V;VEC93(K-@T@:6YY#2!L9&$@*&YO=&5P
M='(I+'D-('-T82!V;VEC93(K-`T-(&QD>"`C-R`[;D]7H%)%4$5!5*!%04-(
MH#@M3D]41:!3150-.FQO;W`@;&1A('9O:6-E,2QX#2!S=&$@=F]I8V4Q*S@L
M>`T@;&1A('9O:6-E,BQX#2!S=&$@=F]I8V4R*S@L>`T@9&5X#2!B<&P@.FQO
M;W`-(&EN>"`[>#TPH"T^H&EN9&5X#2!J;7`@;F]T9610;F4@.V%.1*!$15!!
M4E0-#2HM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM+2TM#2H-*J!A3D2@
M1DE.04Q,6:!42$6@355324.@1$%402Z@H'1(1:!-55-)0Z!$051!H$E3H$1)
M5DE$140-*J!)3E1/H%173Z!465!%4Z!/1J!$051!.J!.3U1%4Z!!3D2@4U!%
M0TE!3*!!0U1)3TY3+@TJH&E&H%1(1:!(24=(H$))5*!)4Z!3150LH%1(14Z@
M5T6@54Y$15)404M%H$&@4U!%0TE!3`TJH$%#5$E/3BR@3U1(15)725-%H%1(
M1:!$051!H$E3H$&@3D]412X-*@TJH&5!0TB@(DY/5$4BH$E3H$%.H$E.1$58
MH$E.5$^@0:!404),1:!/1J!&4D51545.0UF@5D%,5453#2J@4T54H%50H$)9
MH%1(1:!)3DE424%,25I!5$E/3J!23U5424Y%+@TJ#2J@<U!%0TE!3*!!0U1)
M3TY3H$%21:!$15-#4DE"142@55"@04)/5D6@24Z@5$A%H%!!4E-%4@TJH%)/
M551)3D4LH$%.1*!!4D6@5$A)3D=3H$Q)2T6@0TA!3D=%H%9/3%5-12R@1%52
M051)3TXLH$540RX-*J!T2$6@0D%324.@4E5,1:!)4Z!42$%4H$E&H$%.6:!!
M0U1)3TY3H"A%+D<NH%9/3%5-1:!#2$%.1T53*0TJH$%21:!43Z!404M%H%!,
M04-%H$E.H$&@345!4U5212]/3J!!H$Y/5$4LH%1(14Z@5$A%H$1!5$$-*J!#
M3U)215-03TY$24Y'H%1/H%1(3U-%H$%#5$E/3E.@3D5%1%.@5$^@05!014%2
MH$)%1D]21:!42$4-*J!.3U1%H$1!5$$N#2H-*J!T2$6@4$E%0T6@25.@15-3
M14Y424%,3%F@0:!315))15.@3T:@0E)/2T5.H$-(3U)$4RR@04Y$#2J@5$A%
MH$9)4E-4H#(TH$U%05-54D53H$%21:!&3U52+4Y/5$6@0TA/4D13H$)23TM%
M3J!)3E1/#2J@0:!215!%051%1*!3152@3T:@14E'2%2@3D]415,LH%-03$E4
MH$E.5$^@5%=/H$=23U504Z!/1@TJH$9/55*@14%#2"Z@H'1(15)%H$E3H$].
M1:!3152@1D]2H$5!0TB@2$%.1"R@5$A54Z!63TE#1:`Q#2J@04Y$H%9/24-%
MH#*@0D5(059%H%-)34E,05),62Z@H'1(15-%H$%21:!42$6@(C$V+4Y/5$4B
M#2J@345!4U5215.@4D5!1*!)3J!"6:!R96%D,3:@04)/5D4LH$%.1*!42$59
MH$%21:!$149)3D5$#2J@0EF@4U!%0TE&64E.1Z!&25)35*!42$6@1D]54J!.
M3U1%4Z!&3U*@5$A%H%))1TA4H$A!3D0L#2J@5$A%3J!42$6@1D]54J!.3U1%
M4Z!&3U*@5$A%H$Q%1E2@2$%.1"X-*@UN;W1E9&%T80T@9&9B('-E='8Q,B`[
M=D])0T53H#&@04Y$H#(-(&1F8B!S971D=7(-(&1F8B!E:6=H=&@@.VY/5$6@
M1%52051)3TX-(&1F8B!S971V;VPL9@T@.VY/5Z!63TE#12TQH$9/55*@3D]4
M1:!#2$]21`T@.VU%05-54D6@,0T@9&9B(#<R(#MC+38-(&1F8B`V,R`[94(-
M(&1F8B`V,B`[9`T@9&9B(#8P(#MC+34-(#MA3D2@5D])0T4M,J!#2$]21`T@
M9&9B(#0X(#MC+30-(&1F8B`U-2`[9RTT#2!D9F(@-3,@.V8-(&1F8B`U,2`[
M94(-#2`[;45!4U521:`R#2!D9F(@-C@L-C4L-C0L-C`@.V%"H&:@9:!C#2!D
M9F(@-#@L-38L-34L-3,@.V.@84*@9ZT!F#0T@.VU%05-54D6@,PT@9&9B(#<Q
M+#8U+#8S+#8R(#MBH&:@94*@9`T@9&9B(#0X+#4V+#4U+#4S(#MCH&%"H&>@
M9@T-(&1F8B`W,BPV-RPV-2PV,R`[8Z!GH&:@94(-(&1F8B`T."PU,2PU,"PU
M-2`[8Z!E0J!DH&<-#2!D9F(@<V5T=+'-F#2!D9F(@-S4L-C@L-C<L-C,@
M.V5"H&%"H&>@94(-(&1F8B`T."PV,"PU."PU-B`[8Z!CH&)"H&%"#0T@9&9B
M('-E='9O;"QP(#MP24%.3PT@9&9B(#<T+#8V+#8T+#8R(#MDH&8CH&6@9`T@
M9&9B(#0X+#4W+#4U+#4T(#MCH&&@9Z!F(PT-(&1F8B!S971V;VPL<V8-(&1F
M8B`W-"PV-RPV-BPV,B`[9*!GH&8CH&0-(&1F8B`T-BPU."PU-RPU-2`[8D*@
M8D*@8:!G#0T@9&9B('-E='9O;"QP(#MM14%355)%H#@-(&1F8B`W,BPV-"PV
M,BPV,"`[8Z!EH&2@8PT@9&9B(#0V+#4U+#4S+#4R(#MB0J!GH&:@90T-(&1F
M8B!S971V;VPL9@T@9&9B(#<R+#8U+#8T+#8P(#MCH&:@9:!C#2!D9F(@-#0L
M-38L-34L-3,@.V%"H&%"H&>@9@T-(&1F8B`W,"PV-2PV,RPV,B`[8D*@9J!E
M0J!D#2!D9F(@-#0L-3`L-#@L-3,@.V%"H&2@8Z!F#0T@9&9B('-E='9O;"QF
M+3$@.W-405)4H$1%0U)%4T-%3D1/#2!D9F(@-S`L-C<L-C4L-C,@.V)"H&>@
M9J!E0@T@9&9B(#0S+#4Q+#4P+#4U(#MGH&5"H&2@9PT-(&1F8B!S971V;VPL
M9BTR#2!D9F(@-C@L-C<L-C4L-C,@.V%"H&>@9J!E0@T@9&9B(#0X+#4Q+#4P

M+#4V(#MCH&5"H&2@84(-#2!D9F(@<V5T=F]L+'`@.VU%05-54D6@,3,-(&1F
M8B`V."PV,BPV,"PU."`[84*@9*!CH&)"#2!D9F(@-3`L-3,L-3$L-38@.V2@
M9J!E0J!A0@T-(&1F8B`V-RPU."PU-2PV,R`[9Z!B0J!A0J!E0@T@9&9B(#4Q
M+#4U+#4S+#4V(#ME0J!GH&:@84(-#2!D9F(@<V5T=F]L+'`K,2`[0U)%4T,N
M#2!D9F(@-C4L-C`L-3@L-3<@.V:@8Z!B0J!A#2!D9F(@-3$L-3<L-34L-3,@
M.V5"H&&@9Z!F#0T@9&9B('-E='9O;"QP*S(-(&1F8B`V-2PV,BPV,"PU.2`[
M9J!DH&.@8@T@9&9B(#4P+#4S+#4Q+#4V(#MDH&:@94*@84(-#2!D9F(@<V5T
M=F]L+'`K,PT@9&9B(#8U+#8R+#8P+#4Y(#MFH&2@8Z!B#2!D9F(@-#@L-3,L
M-3(L-38@.V.@9J!EH&%"#0T@9&9B('-E='9O;"QP*S0-(&1F8B`V,RPV,"PU
M.2PU-2`[94*@8Z!BH&<-(&1F8B`T."PU,2PU,"PU,R`[8Z!E0J!DH&8-#2!D
M9F(@<V5T=F]L+&8-(&1F8B`U,RPV,RPV,BPV-2`[9J!E0J!DH&8-(&1F8B`T
M-"PT."PT-RPU,"`[84*@8Z!BH&0-#2!D9F(@-30L-C`L-3DL-C,@.V8CH&.@
M8J!E0@T@9&9B(#0U+#4Q+#4P+#0X(#MAH&5"H&2@8PT-(&1F8B!S971V;VPL
M9BLQ#2!D9F(@-C,L-C`L-3DL-34@.V5"H&.@8J!G#2!D9F(@-#,L-3$L-3`L
M-3,@.V>@94*@9*!F#0T@9&9B(#8V+#8P+#4Y+#4W(#MF(Z!CH&*@80T@9&9B
M(#0S+#4Q+#4P+#0X(#MGH&5"H&2@8PT-(&1F8B!S971V;VPL9BLR#2!D9F(@
M-C<L-C`L-3DL-C(@.V>@8Z!BH&0-(&1F8B`T,RPU,2PU,"PU,R`[9Z!E0J!D
MH&8-#2!D9F(@-C@L-C`L-3DL-C(@.V%"H&.@8J!D#2!D9F(@-#,L-3$L-3`L
M-3,@.V>@94*@9*!F#0T@9&9B('-E=&YU;2`[<D5!1*!/3D6@052@0:!424U%
MH$923TV@3D]7H$].#2!D9F(@<V5T=F]L+&8K,2`[1EJ@24:@64]5H%=)3$P-
M(&1F8B!S971D=7(L96EG:'1H*S(@.V&@4T5,1BU)3D153$=%3E2@2$53251!
M5$E/3@T@9&9B(#0S#2!D9F(@<F5S="`[=4Y'051%H%9/24-%H#(-(&1F8B!S
M971V,2`[=D])0T6@,:!/3DQ9#2!D9F(@-#<-(&1F8B!S971D=7(L96EG:'1H
M#2!D9F(@-3`L-3,-(&1F8B!S971V;VPL9B`[<TQ)1TA4H$1%0U)%4T-%3D1/
M#2!D9F(@-38L-3,L-3(L-3,-(&1F8B`U.2PU,RPV,BPU.2PU-BPU,RPU,@T@
M9&9B('-E=&1U<BQE:6=H=&@K,@T@9&9B(#4S#0TJH&1F8J!S971D=7(L96EG
M:'1H*S(-(&1F8B`T,RPT.`T@9&9B('-E=&1U<BQE:6=H=&@-(&1F8B`U,2PU
M-0T@9&9B(#8P+#4U+#4T+#4U#2!D9F(@-C,L-C`L-C<L-C,-(&1F8B`V,"PU
M-BPU-0T@9&9B('-E=&1U<BQE:6=H=&@K,@T@9&9B(#4V#0TJH&1F8J!S971D
M=7(L96EG:'1H*S(-(&1F8B`T,RPT-0T@9&9B('-E=&1U<BQE:6=H=&@-(&1F
M8B`U,RPV,`T@9&9B(#8S+#8P+#8R+#8P#2!D9F(@-C8L-C`L-CDL-C8-(&1F
M8B`V,RPV,"PV,BPV,`T-(&1F8B!S971V,3(-(&1F8B!S971V;VPL9F8@.V9/
M4E1)4U-)34\A#2!D9F(@<F5S=`T@9&9B(#0S(#MV3TE#1:`RH%!%1$%,H%1/
M3D4-(&1F8B!S971V,0T@9&9B('-E=&1U<BQP<F5S=&\@.W!215-43R$-(&1F
M8B`W-"PW,BPW-`T@9&9B(#<U+#<R+#<Q+#<R#2!D9F(@-CDL-S(L-S$L-S(-
M(&1F8B`W-"PW,2PV.2PW,0T-(&1F8B!S971V,3(-(&1F8B`V-RQR97-T("`[
M=D])0T6@,2R@=D])0T6@,@T@9&9B(#<Q+#8R#2!D9F(@-CDL-C`-(&1F8B`W
M,2PV,@T-(&1F8B`W,BPV,R`[8RQE0@T@9&9B(#8Y+#8P(#MA+&,-(&1F8B`V
M-RPU.2`[9RQB#2!D9F(@-CDL-C`@.V$L8PT-(&1F8B`V-BPU-R`[9B,L80T@
M9&9B(#8Y+#8P#2!D9F(@-C<L-3D-(&1F8B`V.2PV,`T-(&1F8B`W,2PV,B`[
M8BQD#2!D9F(@-C<L-3D-(&1F8B`V-BPU-PT@9&9B(#8W+#4Y#0T@9&9B(#8R
M+#4U(#MP4D535$\LH$U%05-54D6@,PT@9&9B(#<Y+#4Y#2!D9F(@-S<L-3<-
M(&1F8B`W.2PU.0T-(&1F8B`X,"PV,`T@9&9B(#<W+#4V#2!D9F(@-S4L-34-
M(&1F8B`W-RPU-@T-(&1F8B`W-"PU,PT@9&9B(#<W+#4W#2!D9F(@-S4L-34-
M(&1F8B`W-RPU-PT-(&1F8B`W.2PU.0T@9&9B(#<U+#4U#2!D9F(@-S0L-3,-
M(&1F8B`W-2PU-0T-(&1F8B`W,BPU,2`[<%)%4U1/H"LT#2!D9F(@-S4L-C<-
M(&1F8B`W-"PV-0T@9&9B(#<U+#8W#0T@9&9B(#<W+#8X#2!D9F(@-S0L-C4-
M(&1F8B`W,BPV,PT@9&9B(#<T+#8U#0T@9&9B(#<Q+#8R#2!D9F(@-S0L-C4-
M(&1F8B`W,BPV,PT@9&9B(#<T+#8U#0T@9&9B(#<U+#8W#2!D9F(@-S(L-C,-
M(&1F8B`W,2PV,@T@9&9B(#<R+#8S#0T@9&9B(#8W+#8P(#MP4D535$^@*S4-
M(&1F8B`W,BPV,PT@9&9B(#<Q+#8R#2!D9F(@-S(L-C,-#2!D9F(@-CDL-C4-
M(&1F8B`W-RPV,@T@9&9B(#<U+#8P#2!D9F(@-S<L-C(-#2!D9F(@-C<L-C,-
M(&1F8B`W-2PV,`T@9&9B(#<T+#4Y#2!D9F(@-S4L-C`-#2!D9F(@-C4L-C(-
M(&1F8B`W-"PU.0T@9&9B(#<R+#4W#2!D9F(@-S0L-3D-#2!D9F(@-C,L-C`@
M.W!215-43Z`K-@T@9&9B(#<R+#8S#2!D9F(@-S$L-C(-(&1F8B`W,BPV,PT-
M(&1F8B`V."PU,PT@9&9B(#8U+#8R#2!D9F(@-C,L-C`-(&1F8B`V-2PV,@T-
M(&1F8B`V-RPU,0T@9&9B(#8S+#8P#2!D9F(@-C(L-3D-(&1F8B`V,RPV,`T-
M(&1F8B`V-2PU,`T@9&9B(#8R+#4Y#2!D9F(@-C`L-3<-(&1F8B`V,BPU.0T-
M(&1F8B!S971V,3(S(#MF3U)-051!H"^@0TA/4D0-(&1F8B!S971D=7(L<')E
M<W1O*S,@.V&@3$E45$Q%H%-,3U=%4@T@9&9B('-E='9O;"QF(#MA3D2@0:!,
M25143$6@4T]&5$52#2!D9F(@<F5S="QR97-T+#0X#2!D9F(@-34L<F5S="QR
M97-T#2!D9F(@<V5T=C$R#2!D9F(@-34L-3@-(&1F8B`V,"QR97-T#2!D9F(@
M<V5T=C$-(&1F8B!S971D=7(L96EG:'1H*C$S(#MH3TQ$H$E4H$]55*!!H$Q)
M5%1,1:!72$E,10T@9&9B(#8T#0T@9&9B('-E=&1U<BQA9&%G:6\O,B`[;D]7
MH%1(1:!!1$%'24^@4E5.#2!D9F(@<V5T=F]L+'`K,B`[<T]&5$52#2!D9F(@
M-C`L-C(L-C0-(&1F8B!S971D=7(L861A9VEO+S0@.S$O-C2@3D]415,-(&1F
M8B`V-2PV-RPV."PW,"PW,BPW,"PV."PV-PT@9&9B('-E=&1U<BQA9&%G:6\-
M(&1F8B`V-0T@9&9B('-E=&1U<BQA9&%G:6\O,BLR#2!D9F(@-C<L-C0-#2!D
M9F(@<V5T=C$R,R`[<T5#3TY$H$9/4DU!5$$O0TA/4D0-(&1F8B!R97-T+')E
M<W0L-#@-(&1F8B`U,RQR97-T+')E<W0-(&1F8B!S971V,3(-(&1F8B`U,RPU
M-@T@9&9B(#8P+')E<W0-(&1F8B!S971V,0T@9&9B('-E=&1U<BQE:6=H=&@J
M,3,-(&1F8B`V-0T-(&1F8B!S971V;VPL<"`[<T]&5$52H%E%5`T@9&9B('-E
M=&1U<BQA9&%G:6\O,B`[<T5#3TY$H&%$04=)3Z!254X-(&1F8B`V-RPV-2PV
M-"PV-2PV-RPV."PV-PT@9&9B('-E=&1U<BQA9&%G:6\O-"`[;4]21:`Q+S8T
MH$Y/5$53#2!D9F(@-C4L-C,L-C(L-C,L-C4L-C(L-C,L-C4-#2!D9F(@<V5T
M9'5R+&5I9VAT:"LQ(#MT2$6@1DE.04R@84Q,14=23Z!314-424].#2!D9F(@
M<V5T=F]L+'`K,B`[8:!,25143$6@3$]51$52+*!IH%1(24Y+#2!D9F(@-3D-
M(&1F8B!S971V,3(S#2!D9F(@-3DL<F5S="PS-@T@9&9B(#4Y+#0W+')E<W0-
M(&1F8B!S971V,3(-(&1F8B`U,"QR97-T#2!D9F(@<V5T=C$-(&1F8B`U,RPU
M-BPU-2PU,RPU.2PU,RPV,BPU,RPU.2PU-BPU-2PU,PT@9&9B('-E='8Q,C,-
M(&1F8B`U,BPT."PS-@T@9&9B(#8Q+')E<W0L<F5S=`T@9&9B('-E='8Q#2!D

M9F(@-3@L-34L-C`L-38L-3,L-38-(&1F8B`U-2PU."PU-2PU,@T@9&9B(#4V
M+#4S+#4P+#4S#2!D9F(@<V5T9'5R+&5I9VAT:"LR#2!D9F(@-3(L-34L-3(L
M-#@-(&1F8B`U,RPU,"PT-RPU,`T-(&1F8B!S971D=7(L96EG:'1H*S,@.VQ%
M3E1/#2!D9F(@<V5T=C$R,PT@9&9B('-E='9O;"QP*S,-(&1F8B!R97-T+')E
M<W0L,S8-(&1F8B!R97-T+#0S+')E<W0-(&1F8B!S971V,3(-(&1F8B!R97-T
M+#0X#2!D9F(@-3`L<F5S=`T@9&9B('-E='8Q#2!D9F(@-3(L-34L-3@L-34-
M(&1F8B`U-BPV,"PV-2PV,@T@9&9B(#8U+#8X#2!D9F(@<V5T9'5R+&5I9VAT
M:"LT(#M2251!4D0-(&1F8B`W,BPW,0T@9&9B('-E=&1U<BQE:6=H=&@K-2`[
M4DE405)$#2!D9F(@-S(L-C<-(&1F8B!S971D=7(L96EG:'1H*S8-(&1F8B`V
M-2PV,@T-(&1F8B!S971D=7(L<')E<W1O*S$-(&1F8B!S971V,3(S#2!D9F(@
M<F5S="QR97-T+#4U#2!D9F(@<F5S="PV,"PU-0T@9&9B('-E=&1U<BPQ-#`-
M(&1F8B`V-"PV,"PU-0T-(&1F8B!R97-T+')E<W0L<F5S=`T@9&9B(')E<W0L
M<F5S="QR97-T#0T@9&9B(')E<W1A<G0@.V%HE.1DE.251530T^[K.N^\I-
ML^V;7LF[K)EN<Y*GNS7/=Y[G55V2K.M:TI*O.S[NO::YRG-79*LZUK2DJ\[/
MNZ]K5LV3;JEN<U=DJSK6M*2KRL^\[NDV\K/O.[I-G.=E=DJSK6M*2KRL^\[N
MDV;ISDV\K/O.[E=DJSK6M*2KSL^[KVFN<E=DJSK6M*2KRL^\[NDV;II5V2K.
MM:TI*F^:I-O*S[SNY79*LZUK2DJ\[/NZ]IK5V2K.M:TI*F^<I-F^;Y-F^:9-
MF^;U=DJSK6M*2IOG:39VFJ39VF^39VG)79*LZUK2DI25*4I.&@````@("TM
M#0````@1$E--`T@("`@("`@("`@("`@("`@("`@P2#-24Y$(,584$%.
M1$E.1R#%6%!%4DE%3D-%#0````@($)9#0``"`@(--415!(14X@S"X@RE5$
M1`T````@("`@4TI51$1`3E=5+D5$50T`````("`M+0T````RE5,62`Q#0``
M`,E.(-1(12#914%2(,]&(,]54B#,3U)$#0`````@,3DY-@T`````("`M+0T@
M#0`BUTA!5"!)3B!42$4@5T]23$0@5T%3(,D@3$]/2TE.1R!!5#\@(-=(050@
M5$A%($A%0TL@25,@64]54@U#3T1%($1/24Y'/R`@R$]7($1/(,D@345%5"!3
M34%25"!!3D0@1D523T-)3U533%D@1T]21T5/55,@5T]-14X@3$E+10U93U4@
M1$_(@T`U$A%($Q!4U0@455%4U1)3TX@R2!#04Y.3U0@04Y35T52+"!"550@
M5$A)4R!,25143$4@5U))5$554"P-04Q/3D<@5TE42"!33TU%(%!%1$%.5$E#
M04Q,62!714Q,+41/0U5-14Y4140@0T]$12P@0T%.($-,14%2(%50(%1(10U&
M25)35"!45T\@R2!42$E.2RX@(-1(25,@5TE,3"!.3U0@0D4@02!615)9($1%
M3E-%(%=2251%55`L($A/3D535"$-S$]/2R!43R!42$4@0T]$12!&3U(@34]2
M12!$151!24PL($%.1"!!3ED@15%5051)3TY3($)%3$]7($-!3B!"10U32TE0
M4$5$(%=)5$A/550@4%)/0DQ%32X-``,E.($-!4T4@64]5($1)1$XG5"!+3D]7
M+"!$24TT(%=!4R!-62!%3E1262!)3E1/(%1(12!214-%3E1,60U(14Q$(#1+
M($1%34\@0T].5$535"X@(,9/4B!-3U)%($E.1D\@3TX@5$A%($-/3E1%4U0L
M($%3(%=%3$P@05,@5$A%#4]42$52($5.5%))15,@*#$W($5.5%))15,@24X@
M04Q,*2P@4T5%2R!912!42$4@Q%))5D5.($A/344@4$%'12!!5`U(5%10.B\O
M4T](3RY)3U,N0T]-+R4W14-/3TQ(3D0O#2`-`,9)4E-4+"!615)9($)2245&
M3%DL(%1(12!+15E04D534T53($A!5D4@5$A%($9/3$Q/5TE.1R!!0U1)3TY3
M.@T-`#0@("`M+2#455)"3R!-3T1%#0#$("`@+2T@SD]234%,($U/1$4@*#3$
M("L@,\0@4D]4051)3TY3+"!!3D0@3DE#12!!3D0@0T%354%,*0T`QC0@("TM
M(#3$+4U/1$4N("#!3$P@(C/$(B!23U1!5$E/3E,@05)%($A!3%1%1`T`TB_3
M("TM(#/$+4U/1$4N("#!3$P@(C3$(B!23U1!5$E/3E,@05)%($A!3%1%1`T`
M+B`@("TM(,1/5%1%1"!,24Y%(%1/1T=,10T`TU!!0T4@041604Y#15,@5$\@
M5$A%($Y%6%0@3T)*14-4+@T-`-1(12!#3T1%($E3(#0P.34@0EE415,@3$].
M1RP@04Y$(%=!4R!!(%)/64%,(%!!24X@5$\@1T54#5=/4DM)3D<@049415(@
M0T]-4%)%4U-)3TXN("#42$4@355324,@25,@T%)%3%5$12`C,B!&4D]-(-1(
M12#714Q,+0W414U015)%1"#+3$%62452($)9(,HNTRX@PD%#2"X@(,D@0D]2
M4D]7140@*$%.1"!)35!23U9%1"D@5$A%($Q)3D4-1%)!5TE.1R!23U5424Y%
M($923TT@5$A%($-50D4S1"!04D]'4D%-4R!!3D0@4U1/3$4@5$A%(%!!5%1%
M4DY3($]55`U/1B#03TQ91T].04U9+"!/5$A%4E=)4T4@5$A%($-/1$4@25,@
M5U))5%1%3B!&4D]-(%-#4D%40T@L($E.0TQ51$E.1PU42$4@355324,@4D]5
M5$E.12X@(-1(050@0U)!6ED@5$A)4D0@3T)*14-4($A!4R!&3U525$5%3B!3
M241%4R!)3B`SQ"P-04Y$(%1(12`TQ"!/0DI%0U0@04Q/3D4@2$%3(#,R(%!/
M24Y44R!7251((#DV($Q)3D53($-/3DY%0U1)3D<@5$A%#5!/24Y44RP@4T\@
M5T5,3"!/5D52(#$P,"!,24Y%4R!!4D4@0D5)3D<@1%)!5TX@050@02!424U%
M+B`@R2!705,-4T]214Q9(%1%35!4140@5$\@4%54($$@1$]#2TE.1R!"05D@
M3TX@3TY%($]&(%1(12!3241%4R!/1B!42$4@,\0-1U59("A!($Q)5%1,12`B
MQ4Q)5$4B($A534]2(%1(15)%*2!"550@4D%.($]55"!/1B!424U%($%.1"!2
M3T]-+@W!1E1%4B!$14-/35!215-324].(%1(12!#3T1%(584%.1%,@02!,
M25143$4@0DE4+"!!3D0@24X@5$A%($5.1"!)5`U,14%615,@5$A%(#A+($)%
M5%=%14X@)#0P,#`M)#8P,#`@1E)%12P@0E54(%5315,@4%)%5%19($U50T@@
M159%4EE42$E.1PU%3%-%+@T`U$A%($9)4E-4($]"2D5#5"!)4R!!(#3$($-5
M0D4L($]&5$5.($-!3$Q%1"!!($A94$520U5"12X@(-E/50U#04X@4T5%($$@
M4TU!3$P@0U5"12!)3E-)1$4@3T8@04Y$($-/3DY%0U1%1"!43R!!($Q!4D=%
M4B!#54)%+B`@R48@64]5#4Q/3TL@02!,25143$4@0TQ/4T52+"!93U4@34%9
M($Y/5$E#12!42$%4($E.+4)%5%=%14X@5$A%(%173R!#54)%4R!!4D4-4T]-
M12!-3U)%($-50D53+B`@*-=(14X@64]5(%-,24-%($$@,\0@0U5"12P@64]5
M($=%5"!!(#+$($-50D4@+2T@02`-4U%505)%+B`@UTA%3B!93U4@5$%+12!!
M(%-,24-%($]&(%1(12!(65!%4D-50D4L(%E/52!'150@02`SQ"!#54)%*2X@
M(,%3#4E4(%)/5$%415,@04Q/3D<@2513($9/55)42"!#3T]21$E.051%+"!4
M2$4@0U5"12!&3TQ$4R!)3B!54$].#4E44T5,1BX@(,].12!705D@5$\@3$]/
M2R!!5"!)5"!)4R!42$%4(%1(12!#54)%4R!35$%25"!43R!#2$%.1T4-4$]3
M251)3TY3("TM($%&5$52(#$X,"!$14=21453($]&(%)/5$%424].(%1(12!)
M3E-)1$4@0U5"12!)4R!/3B!42$4-3U544TE$12!!3D0@5$A%($]55%-)1$4@
M0U5"12!)4R!/3B!42$4@24Y3241%+B`@U$A%($A94$520U5"12!(05,-3$E4
M15)!3$Q9(%154DY%1"!)3E-)1$4M3U54+@T`U$A%(%!23T=204T@5T]22U,@
M1DE.12!)3B#0P<P@04Y$(,[4T\,L($%,5$A/54=((-#!S"!&3TQ+4PU724Q,
M($=%5"!42$4@5%5.12!03$%924Y'($%4(%1(12!74D].1R!34$5%1"!!3D0@
M5%)!3E-03U-%1"!)3E1/#4$@1$E&1D5214Y4($M%62X-`,](%E%4RP@3TY%
M(%1(24Y'(,D@4D5!3$Q9($Q)2T4@25,@5$A%($)!0TM'4D]53D0@3TX@5$A%
M(%-%0T].1`U/0DI%0U0@+2T@3TX@35D@,3`X-"!)5"!,3T]+4R!,24M%(%)/

```
M4$4N("#42$E3($E3($$@0T].4T51545.0T4@3T8@5$A%#5=!62#6R<,@1T5.
M15)!5$53($-/3$]24R`M+2!%6%1202!#3TQ/4E,@3U544TE$12!/1B!42$4@
M3D]234%,(#$V($%210U"14E.1R!'14Y%4D%4140L($)%0T%54T4@5%=/($A)
M4D53($-/3$]24R!!4D4@0D5)3D<@4$Q!0T5$($Y%6%0@5$\-14%#2"!/5$A%
M4BX@(,E&(%E/52!,3T]+($%4($E4($].($$@0DQ!0TL@04Y$(%=(251%($U/
M3DE43U(L(%E/52!724Q,#4I54U0@4T5%(%1(24-+($1)04=/3D%,($Q)3D53
M+B`@U$A)4R!615)9($U50T@@4U524%))4T5$($U%(%=(14X@R2!&25)35`U3
M05<@250A("#&24Y$(%1(12#-05)#2"`Q.3@U(,G%Q<4@TU!%0U1254T@05)4
M24-,12!&3U(@34]212!)3D9/4DU!5$E/3@U/3B!72%D@ULG#($)%2$%615,@
M5$A)4R!705DN#0#&24Y!3$Q9+"!93U4@34%9($Y/5$E#12!33TU%($Q)5%1,
M12!'3$E40TA%4R!&4D]-(%1)344@5$\@5$E-10U)3B!$4D%724Y'(%1(12`T
MQ"!/0DI%0U13+B`@U$A!5"!)4R!-62!3049%5%D@5D%,5D4@04Y$($M%15!3
M(%1(12!04D]'4D%-#4923TT@3$E415)!3$Q9($1%4U123UE)3D<@251314Q&
M+"!)3B!33TU%5$E-15,@4U!%0U1!0U5,05(@1D%32$E/3BX@(,](#5=%3$PN
M#0W!(,A!3D19(,=,3U-305)9#2TM+2TM+2TM+2TM+2TM+2T-`-!/3%E'3TXZ
M(,$@4D5#5$E,24Y%05(@0TQ/4T5$(%!,04Y%($9)1U5212!/1B!!3ED@3E5-
M0D52($]&(%-)1$53+@T`+2TM+2TM+0T-`-9%0U1/4CH@P2!$25)%0U1%1"!,
M24Y%(%-%1TU%3E0@2$%624Y'($U!1TY)5%5$12!!3D0@1$E214-424].+@T@
M("`@("`@("TM+2TM+0T-`,D@1$\@3D]4($M.3U<@2$]7(%1(12!415)-(")&
M24Q,140@5D5#5$]2(B!#04U%($E.5$\@5D]'544L($)55`U)5"!)4R!-14%.
M24Y'3$534RP@3D]4(%1/($U%3E1)3TX@02!,25143$4@4TE,3%D@+2T@5TA!
M5"!73U5,1"!!3@TB54Y&24Q,140@5D5#5$]2(B!,3T]+($Q)2T4L(%173R!0
M3TE.5%,@5TE42"!!3B!!4E)/5R!!5"!/3D4@14Y$/R`@STY%#4U!62!!4R!7
M14Q,(%1!3$L@04)/550@1DE,3$5$($Q)3D53($%.1"!&24Q,140@4$])3E13
M+@T-`-1(55,L(,D@4$Q%040@5TE42"!42$4@0T]-355.2519(%1/($Y/5"!2
M149%4B!43R!03TQ91T].4R!!4PU614-43U)3($%.1"!&24Q,140@4$],64=/
M3E,@05,@1DE,3$5$(%9%0U1/4E,N("#03TQ91T].4R!.145$(%E/55(-2$5,
M4"P@04Y$($A!5D4@0D5%3B!$25-#4DE-24Y!5$5$($%'04E.4U0@1D]2(%1/
M3R!,3TY'($Y/5RX@(,I54U0@3TY%#5--04Q,($1/3D%424].($].(%E/55(@
M4$%25"!/1B!!($-/4E)%0U0@34%42$5-051)0T%,(%)%1D5214Y#12!#04X-
M2$5,4"!3059%(%1(12!,259%4R!/1B!/3D4L(%1%3BP@159%3B!(54Y$4D5$
M4R!/1B!03TQ91T].4RP@0D]42"!!0E)/040-04Y$($A%4D4@050@2$]-12X@
M(,E.1$E6241504Q3(%=!3E1)3D<@5$\@0T].5%))0E5412!-3U)%($U!62!3
M4$].4T]2#4E.1$E6241504P@4$],64=/3E,[($$@2TE4(%=)3$P@0D4@4T5.
M5"!43R!93U4@0T].5$%)3DE.1R!42$4@3D%-10U/1B!42$4@4$],64=/3B!!
M3D0@050@4D5'54Q!4B!)3E1%4E9!3%,@02!024-455)%($]&(%1(12!03TQ9
M1T].(%=)3$P@0D4-4T5.5"!43R!93U4L(%-/(%E/52!-05D@34].251/4B!4
M2$4@4%)/1U)%4U,@3T8@64]54B!005)424-53$%2(%!/3%E'3TXN#=-/344@
M4$],64=/3E,@05)%($-214%4140@54Y#3$]3140L($%.1"!33TU%($1/($Y/
M5"!'150@5$A%($Y%0T534T%260U)3DL@3U(@4%)/1U)!34U)3D<@4TM)3$P@
M5$\@4%)/4$523%D@1DE,3"!42$5-+"!"550@0D4@250@02!154%$3$E,051%
M4D%,#4]2($1%0T%'3TXL(%1205!%6DE532!/4B!005)!3$Q%3$]'4D%-+"!7
M251((%E/55(@2$5,4"!712!#04X@159%3E1504Q,60U-04M%($%,3"!03TQ9
M1T].4R!#3$]3140@04Y$($953$PL($9/4B!!($)%5%1%4BP@34]212!#259)
M3$E:140@5T]23$0N#=1(04Y+(%E/52!&3U(@64]54B!424U%+"!!3D0@QT]$
M($),15-3($%,3"!42$4@3$E45$Q%($=%3TU%5%))0T%,#4-/3E-44E5#5$E/
M3E,L($Y/($U!5%1%4B!42$5)4B!$24U%3E-)3TX@3U(@0T].1DE'55)!5$E/
M3BX-(`T@#=1(12#)1$5!#2TM+2TM+2TM(`T`U$A)4R!04D]'4D%-($1)4U!,
M05E3($$@4D504D5314Y4051)3TX@3T8@4T]-12!&3U52+41)345.4TE/3D%,
M#4]"2D5#5%,@+2T@1D]54B`TQ"!/0DI%0U13+"!!4R!!($U!5%1%4B!/1B!&
M04-4+"!%04-(($].12!/1B!42$5-#4$@-,0@04Y!3$]'($]&($$@5$A2144M
M1$E-14Y324].04P@3T)*14-4+B`@Q4%#2"!30U)%14X@0T].5$%)3E,@1D]5
M4@U364U-151262U214Q!5$5$(#/$($]"2D5#5%,@04Y$($].12`TQ"!!3D%,
M3T<@3T8@5$A%($]"2D5#5"P@4D]4051%1`U!3D0@4%)/2D5#5$5$($923TT@
M-,0@24Y43R`RQ"X-#0#43R!$15-#4DE"12!42$4@1D]54BU$24U%3E-)3TY!
M3"!/0DI%0U13($E3($Y/5"!33R!43U5'2"X-U$A%(#3$($-50D4@*%1(12!(
M65!%4D-50D4I($E3(%1(12!&25)35"!43R!"12!$25-03$%9140L($%.1"!)
M5"!)4PU42$4@4U1!4E1)3D<@4$])3E0@1D]2(%1(12!,051%4B!/0DI%0U13
M+"!!3D0@250@25,@04Q33R#)(%1(24Y+(%1(10U%05-)15-4(%1/(%-%12!7
M2$%4($E3($=/24Y'($].(%=)5$@N("#42$5212!)4R!.3U1(24Y'(%)%04Q,
M62!34$5#24%,#4%"3U54($9/55(@1$E-14Y324].4R`M+2!7251(($$@,\0@
M3T)*14-4($5!0T@@4$])3E0@25,@1$5&24Y%1"!"62!42%)%10U#3T]21$E.
M051%4RP@4T%9("A8+%DL6BDN("#!(#3$(%!/24Y4($A!4R!&3U52($-/3U)$
M24Y!5$53+"!305D-*%<L6"Q9+%HI+B`@U$A%(#/$($-50D4@2$%3($5)1TA4
M(%9%4E1)0T53($%4#0T`*"LO+3$L("LO+3$L("LO+3$I#0W42$52149/4D4@
M02!615)9($Y!5%5204P@15A414Y324].($E.5$\@1D]54B!$24U%3E-)3TY3
M(%=/54Q$($)%#0T`*"LO+3$L("LO+3$L("LO+3$L("LO+3$I#0W&3U(@02!4
M3U1!3"!/1B!325A4145.(%9%4E1)0T53+B`@U$\@3$]/2R!!5"!)5"!!3D]4
M2$52(%=!63H-#0`H,2P@*R\M,2P@*R\M,2D-`"@M,2PK+RTQ+"`K
M+RTQ+"`K+RTQ*0T-U$A!5"!)4RP@050@5STQ(%=%($=%5"!!($-50D4L($%.
M1"!!5"!7/2TQ(%=%($=%5"!!3D]42$52($-50D4N("#)3@U&04-4+"!)1B!7
M12!404M%($$@(E-,24-%(B!/1B!/55(@2%E015)#54)%+"!712!'150@02`S
MQ"!#54)%+@W#3TU!4$%212!43R!404M)3D<@02!33$E#12!/1B!!(#/$($-5
M0D4L(%=(15)%(%E/52!'150@02!3455!4D4-*$$@,L0@0U5"12P@248@64]5
M(%=)3$PI+@T`U$A)4R!)4R!$14U/3E-44D%4140@5TA%3B!42$4@0T]$12!&
M25)35"!35$%25%,@55`@+2T@5$A%(%!23T=204T-(D=23U=3(B!!($-50D4@
M1E)/32`PQ"`M/B`QQ"`M/B`RQ"`M/B`SQ"`M/B`TQ"X@(,%4(%1(12`TQ"!3
M5$%'10U42$5212!)4R!!(%--04Q,15(@0U5"12!)3E-)1$4@3T8@02!,05)'
M15(@0U5"12P@5TE42"!#54)%4R!)3BU"1517145.#51(12!45T\N("`HR48@
M64]5($%212!#55))3U53($%3(%1/($A/5R#)($1)1"!42$4@(D=23U=)3D<B
M+"!3144@5$A%#4-/1$4@1$530U))4%1)3TX@0D5,3U<@1D]2($$@1D57($1%
```

M5$%)3%,I+@T`SD585"P@05,@5$A%($-50D4@0D5'24Y3(%1/(%)/5$%412P@
M250@(D9/3$13($E.(B!/3B!)5%-%3$8-*$]2+"!)1B!93U4@3$E+12P@250@
M54Y&3TQ$4R$I+B`@TD]4051)3TY3($%212!.3R!$249&15)%3E0@5$A!3@U4
M2$59($A!5D4@04Q705E3($)%14XN("#43R!$3R!!(#/$(%)/5$%424].+"!2
M14-!3$P@5$A!5"!42$4@3T)*14-4($E3#5)/5$%4140@24X@5$A%(%@M62!0
M3$%.12P@5$A%(%DM6B!03$%.12P@04Y$(%1(12!8+5H@4$Q!3D4N("#43R!2
M3U1!5$4-24X@5$A%(%@M62!03$%.12!"62!!3B!!3D=,12!02$DZ#0T`6$Y%
M5R`](%@J0T]3*%!(22D@+2!9*E-)3BA02$DI#0!93D57(#T@6"I324XH4$A)
M*2`K(%DJ0T]3*%!(22D-#==%3$PL($%.62!45T\@0T]/4D1)3D%415,@1D]2
M32!!(%!,04Y%+"!33R!)3B!&3U52($1)345.4TE/3E,@5$A%4D4-05)%($I5
M4U0@5%=)0T4@05,@34%.62!03$%.15,@5$\@4D]4051%($E.+B`@R4X@4$%2
M5$E#54Q!4BP@5$A%#5!23T=204T@1$]%4R!23U1!5$E/3E,@24X@5$A%(%53
M54%,(%!,04Y%4R`H6"U9+"!9+5HL(%@M6BD@04Y$#4%,4T\@1$]%4R!!(%-)
M3D=,12!23U1!5$E/3B!)3B!42$4@5RU8(%!,04Y%+"!42$%4($E3+`T-`%=.
M15<@/2!7*D-/4RA02$DI("T@6"I324XH4$A)*0T`6$Y%5R`](%<J4TE.*%!(
M22D@*R!8*D-/4RA02$DI#0W)($1)1$XG5"!&145,($%.62!'4D5!5"!.145$
M(%1/(%)/5$%412!42%)/54=(($585%)!(%!,04Y%4R!)3E9/3%9)3D<-5$A%
M(%<M0T]/4D1)3D%412`H5$A%(%<M62!!3D0@5RU:(%!,04Y%4RDN("#72$5.
M(%!(23TY,"!$14=21453+`U/4B`Q.#`@1$5'4D5%4RP@3D]424-%(%1(050@
M5$A%($-/3U)$24Y!5$53(%12041%(%!,04-%4RP@5$A%3B!'3PU43R!42$5)
M4B!.14=!5$E615,N("#42$E3($U%04Y3(%1(050@05,@4$A)($E3($E.0U)%
M05-%1"P@24X@15-314Y#10U42$4@24Y.15(@04Y$($]55$52($-50D53($%2
M12!'3TE.1R!43R!#2$%.1T4@4$]3251)3TY3+"!!3D0@5$A)4PU42$5.($58
M4$Q!24Y3(%1(12!53D9/3$1)3D<@5$A!5"!)4R!3145.($].(%1(12!30U)%
M14XN#0#42$4@TB_3($M%62!'3T53($E.5$\@,\0@34]$12!"62!:15)/24Y'
M($]55"!42$4@04Y'3$4-24Y#4D5-14Y4($9/4B!42$4@5RU8(%!,04Y%+B`@
MMR4X@149&14-4+"!42$4@-,0@4D]4051)3TX@25,@1E)/6D5.+@W42$4@QC0@
M2T59(%I%4D]3($]55"!42$4@6"U9+"!9+5HL($%.1"!8+5H@04Y'3$4@24Y#
M4D5-14Y44RP@3$5!5DE.1PU/3DQ9(%1(12!7+5@@4D]4051)3TXN("#&-"!&
M3TQ,3U=%1"!"62#2+],@5TE,3"!42$52149/4D4@1E)%15I%#51(12!)34%'
M12!#3TU03$5414Q9("TM(%5312#$($]2(#0@5$\@1T54($E4($=/24Y'($%'
M04E.+@T`U$A%4D4@25,@4U1)3$P@5$A%($E34U5%($]&(%9)4U5!3$E:24Y'
M($$@-,0@3T)*14-4+B`@U$A)4PU32$]53$0@3D]4($)%(%-54E!225-)3D<@
M+2T@049415(@04Q,+"!712!(059%($%,3"!3145.(#/$($]"2D5#5%,-1%)!
M5TX@3TX@02`RQ"!#3TU0551%4B!30U)%14X@*$]2($$@,L0@4$E%0T4@3T8@
M4$%015(I+B`@R48@5T4@0T%.#4=%5"!&4D]-(#/$(%1/(#+$(%1(14X@5T4@
M3U5'2%0@5$\@0D4@04),12!43R!'150@1E)/32`TQ"!43R`SQ`TH04Y$($92
M3TT@5$A%4D4@24Y43R`RQ"DN("#214-!3$P@5$A!5"!!(#/$(%!23TI%0U1)
M3TX@1%)!5U,@02!,24=(5`U205D@1E)/32!42$4@3T)*14-4+"!42%)/54=(
M($$@3$E45$Q%(%!)3DA/3$4@3$]#051%1"!!5"!42$4@3U))1TE.+`U!3D0@
M1DE.1%,@5$A%($E.5$524T5#5$E/3B!7251(($$@4$E%0T4@3T8@1DE,32!,
M3T-!5$5$($%4(%H]1"P@00U#3TY35$%.5#H-#0#,(#T@5"`J("A8,2Q9,2Q:
M,2D@25,@35D@3$E'2%0@4D%9+"!33R!4/40O6C$@1TE615,@5$A%#0``"`@
M($E.5$524T5#5$E/3B!7251((%1(12!&24Q-($]&($$@4D%9($923TT-````
M("`@5$A%(%!/24Y4("A8,2Q9,2Q:,2D@4$%34TE.1R!42%)/54=((%1(10T`
M```@("!/4DE'24XN#0W33R!42$E3($E3(%9%4ED@14%362!43R!%6%1%3D0@
M24Y43R`TQ"`M+2!324U03%D@4%)/2D5#5"!&4D]-(#3$(`U)3E1/(#/$(%1(
M4D]51T@@5$A%($]224=)3CH-#0#,(#T@5"`J("A7,2Q8,2Q9,2Q:,2D@($Q%
M5"!4/40O5S$-#0`M/B#,,R`]("A$+"!$+U<Q("H@6#$L($0O5S$@*B!9,2P@
M1"]7,2`J(%HQ*0T-U$A%(%@L62Q:($-/3U)$24Y!5$53($%212!42$5.(%!2
M3TI%0U1%1"!&4D]-(#/$($E.5$\@,L0L($%'04E.#51(4D]51T@@5$A%($]2
M24=)3BX@(-1(25,@1TE615,@02`B4$524U!%0U1)5D4B(%9)15<@3T8@5$A%
M(#3$#4]"2D5#5"X-`,Y/5RP@5TA!5"!)4R!42$4@-,0@04Y!3$]'($]&($$@
M5$544D%(14123TXL($]2($%.($]#5$%(14123TX_#<D@4D5!4T].140@5$A%
M32!/550@0ED@5%)924Y'(%1/(%1(24Y+($]&(%=(050@,\0@3T)*14-44R#)
M($-/54Q$#41%4DE612!35$%25$E.1R!&4D]-($$@0U5"12X@(-1(050@25,L
M(%1!2TE.1R!!($-50D4L($%.1"!#551424Y'#4%705D@4$E%0T53($]&($E4
M+B`@QD]2($E.4U1!3D-%+"!43R!$3R!42$4@,30M4TE$140@1U59+"!324U0
M3%D-5$%+12!42$4@34E4])3E0@3T8@14%#2"!,24Y%(%-%1TU%3E0@3TX@
M5$A%($-50D4@+2T@5$A)4R!(05,-5$A%($5&1D5#5"!/1B!#551424Y'($]&
M1B!42$4@0T]23D524R!/1B!42$4@0U5"12X@(,)9($1%1DE.24Y'#51(24Y'
M4R!)3B!42$E3(%=!62P@250@25,@1D%)4DQ9(%-44D%)1TA41D]25T%21"!4
M3R!%6%1%3D0@5$A%($]"2D5#5%,-24Y43R!&3U52($1)345.4TE/3E,N("`H
MMR2!705,@2$%04$E%4U0@5$\@4D5!3$E:12!(3U<@5$\@1$\@02!4151204A%
M1%)/3BDN#=-%12!42$4@1DE,12!/0DI%0U13+E,@1D]2($U/4D4@1$5404E,
M4R!/3B!42$4@24Y$259)1%5!3"!/0DI%0U13+@W.05154D%,3%D@14%#2"!(
M05,@4T]-12!324U)3$%22519(%1/(%1(12!#54)%.B!42$5212!)4R!!3B!)
M3DY%4B!/0DI%0U0-*$4N1RX@02!4151204A%1%)/3BD@04Y$($%.($]55$52
M+B`@U$A%(%173R!!4D4@0T].3D5#5$5+"!!3D0@14%#2`U3150@3T8@0T].
M3D5#5$E/3E,@1D]235,@04Y/5$A%4B!/0DI%0U0L(%-/(%1(050L($9/4B!)
M3E-404Y#12P@5$A%4D4-05)%(%1%5%)!2$5$4D].4R!)3BU"1517145.(%1(
M12!)3DY%4B!!3D0@3U5415(@5$544D%(14123TY3+@T`QDE.04Q,62P@5$\@
M2$5,4"!)3B!625-53$%)6DE.1R!42$4@3T)*14-44R#)(%-454-+($$@1$]4
M5$5$#4Q)3D4@0T%004))3$E462!)3BX@(-1(12!$3U14140@3$E.15,@24X@
M1T5.15)!3"!#3TY.14-4(%1(12`B24Y.15(B#4%.1"`B3U5415(B(#/$($]"
M2D5#5%,@+2T@5%523DE.1R!42$5-($]&1B!,1513(%E/52!42$5.(%-%12!4
M2$4-5%=/($]"2D5#5%,@24Y415)!0U0N("`HU$A%(%1(25)$($]"2D5#5"!7
M05,@34E'2%19($E-4%)%4U-)5D4M3$]/2TE.1PU"149/4D4@R2!!1$1%1"!4
M2$5312!'55E3(2`Z*0T@``W42$4@PT]$10TM+2TM+2TM+0T`SD]7+"!)5"!)
M4R!-62!#3TY3241%4D5$($]024Y)3TX@5$A!5"!42$4@0T]$12!)4R!!5T95
M3$Q9#5=%3$P@1$]#54U%3E1%1"P@4T\@5$A%4D4@25-.)U0@5$]/($U50T@@

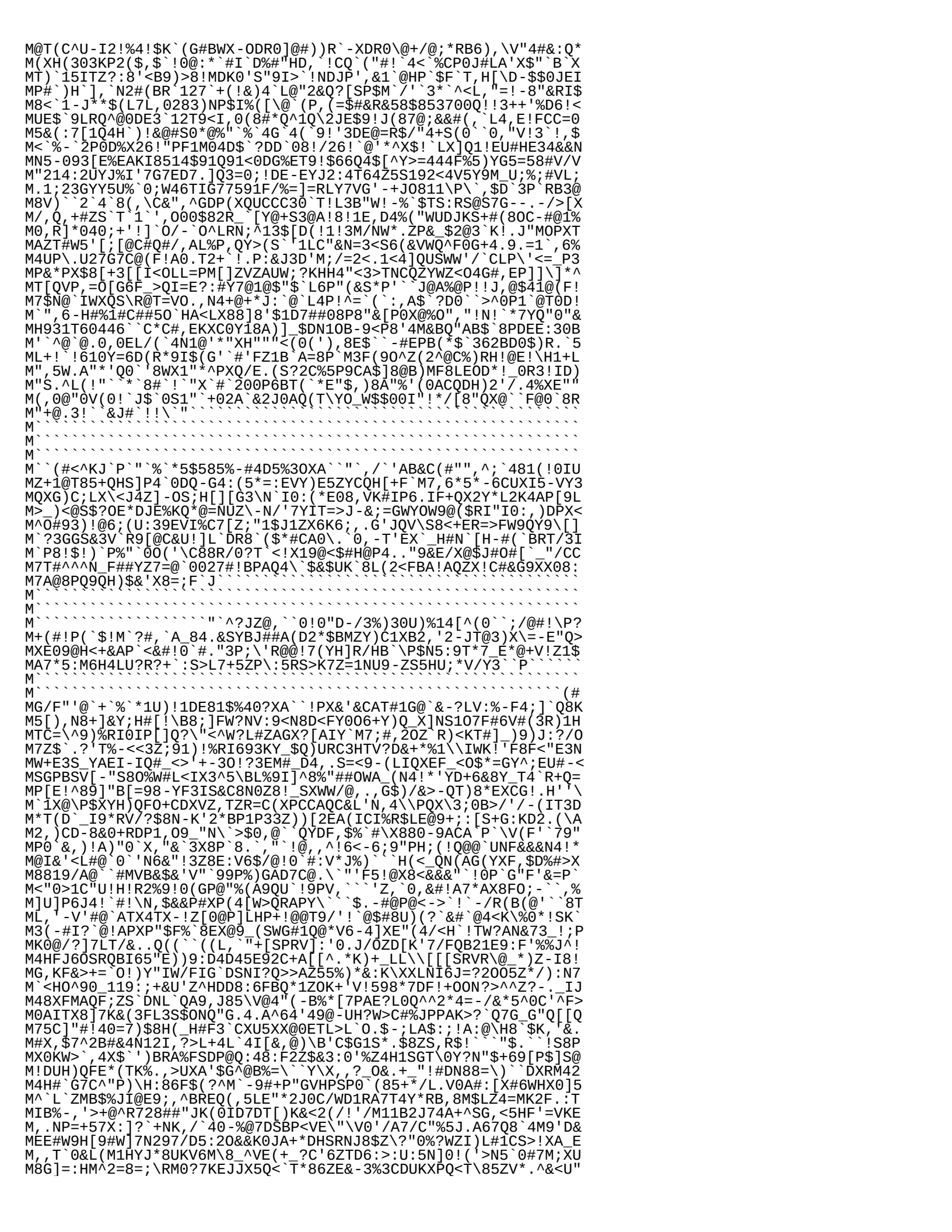
M5$\@4T%9+"!"550@02!&15<@1T5.15)!3`U42$E.1U,@05)%(%=/4E1(($U%
M3E1)3TY)3D<N#0`BQU)/5TE.1R(@5$A%(%!/24Y44R!)4R!214%,3%D@14%3
M62`M+2!324U03%D@4U1!4E0-14%#2"!#3T]21$E.051%($%4(%I%4D\L($%.
M1"!'4D%$54%,3%D@24Y#4D5!4T4@250@3U54(%1/($E44R!&24Y!3`U604Q5
M12X@(,)9($1/24Y'(%1(25,@1DE24U0@5TE42"!42$4@6"U#3T]21$E.051%
M4RP@5$A%3B!42$4@62U#3T]21%,L#51(14X@6BP@5$A%3B!7+"!42$4@0U5"
M12!'4D]74R!!($1)345.4TE/3B!!5"!%04-((%-415`N("#)($1/3B=4#41/
M($%.651(24Y'($9!3D-9(%=)5$@@5$A%($]42$52($]"2D5#5%,@+2T@04Q,
M($-/3U)$24Y!5$53($%210U'4D]73B!%455!3$Q9+"!33R!42$4@3T)*14-4
M4R!'4D]7($]55%=!4D13($923TT@5$A%($]224=)3B`H05,@3U!03U-%1`U4
M3R!33TU%(%-/4E0@3T8@6D]/32!%1D9%0U0I+@T`Q4%#2"`TQ"!#2$%204-4
M15(@25,@02`Q,E@Q,B!#2$%204-415(@1U))1"P@5TA)0T@@1TE615,@02`Y
M-E@Y-@U025A%3"!$4D%724Y'($%214$L($%.1"!404M%4R!54"!42$4@1DE2
M4U0@,30T($-(05)!0U1%4E,N("#%04-(#3/$($-(05)!0U1%4B!54T53($$@
M-5@U($-(05)!0U1%4B!'4DE$+"!'259)3D<@-#!8-#`L($%.1"!404M)3D<-
M55`@5$A%($Y%6%0@-"HR-3TQ,#`@0TA!4D%#5$524RP@1D]2($$@5$]404P@
M3T8@,C0T(%-/($9!4BX@(,E.#45)1TA4($]&(%1(12!214U!24Y)3D<@,3(@
M0TA!4D%#5$524R!!4D4@1D]54B!0051415).4R!!3D0@5$A%25(-Q<_2(",D
MQL8@0T]-4$Q%345.5%,L(%=(24-(($%212!54T5$($E.(%1(12!"04-+1U)/
M54Y$(%1)3$E.1U,@04Y$#4%212!54T5$($E.1$E214-43%D@24X@5$A%(%!!
M5%1%4DX@1DE,3%,N#0#324Y#12!42$4@1DE.04P@6"U9($-/3U)$24Y!5$53
M($-!3B!204Y'12!&4D]-("TT."XN-#@L(%1(25,-4$Q!0T53($$@4D535%))
M0U1)3TX@3TX@5$A%($E.251)04P@5D%,5453($9/4B!42$4@0T]/4D1)3D%4
M15,N("#&3U(-4%524$]315,@3T8@04-#55)!0UD@04Y$(%-50T@@0T]/4D1)
M3D%415,@35535"!/1B!#3U524T4@0D4@4T-!3$5$+`U33R!42$%4(%=(24Q%
M($$@0T]/4D1)3D%412!,24M%("@Q+#$L,2PQ*2!)4R!#3TY614Y)14Y4($9/
M4B!42$E.2TE.1RP-02!#3T]21$E.051%($Q)2T4@*#$V+#$V+#$V+#$V*2!)
M4R!-54-(($)%5%1%4B!354E4140@5$\@5$A%#4E-4$Q%345.5$%424].("TM
M(%1(050@25,L(%1(12!/4DE'24Y!3"!#3T]21$E.051%(%-#04Q%1"!"62!!
M($9!0U1/4@U/1B!325A4145.($]2(%-/+B`@U$A%(%1!0DQ%(%)!3D=%(%)%
M4U1224-44R!42$E3(%-#04Q)3D<@1D%#5$]2.B!42$4--,0@0T]/4D1)3D%4
M12!7251(($Q!4D=%4U0@3$5.1U1((%1(050@R2!54T4@25,@*#$L,2PQ+#$I
M+"!72$E#2`U(05,@3$5.1U1((#(N("#42%53+"!!1E1%4B!23U1!5$E/3BP@
M250@25,@4$]34TE"3$4@5$A!5"!)5"!724Q,($Q)10U/3B!!3B!!6$E3(%=)
M5$@@0T]/4D1)3D%412P@4T%9("@R+#`L,"PP*2X@(--)3D-%($-/3U)$24Y!
M5$53($U54U0-3D]4($580T5%1"`T."!)3B!42$4@24U03$5-14Y4051)3TXL
M(%1(25,@4U5'1T535%,@02!30T%,24Y'($9!0U1/4@U/1B`R-"X-`,%3($$@
M4%)!0U1)0T%,(%!/24Y4+"!42$4@4$])3E13($Y%5D52(%)%04Q,62!(250@
M5$A)4PU-05A)355-+"!33R!)3B!04DE.0TE03$4@02!,05)'15(@4T-!3$E.
M1R!&04-43U(@0T]53$0@0D4@55-%1"X-P4Q415).051)5D5,62!42$4@4%)/
M2D5#5$E/3B!23U5424Y%($-!3B!024-+(%50(%1(12!33$%#2RP@5TA)0T@@
M25,-5TA!5"!$24TT(%5315,N#0#42$4@1DE24U0@4TU!4E0@5$A)3D<@R2!$
M240@5T%3(%1/($1)5$-((%1(12!/3$0@34542$]$#4]&($-/35!55$E.1R!2
M3U1!5$E/3E,N("#)3E-414%$($]&($-!3-53%424Y'($$@0DE'(%)/5$%4
M24].#4U!5%))6"P@R2!#04Q#54Q!5$4@4T]-12!"24<@5$%"3$53($]&($8@
M6"`H4RD@/2!8*E-)3BA3*2P@04Y$($Q%5`U42$4@04Y'3$4@4R!204Y'12!&
M4D]-(#`N+C$R-RX@(-1/($=%5"!!(%1!0DQ%($]&($-/4RA3*2#)(%-)35!,
M60U015))3T1)0T%,3%D@15A414Y$(%1(12!324Y%(%1!0DQ%($)9($-/4%E)
M3D<@5$A%($9)4E-4(#,R($)95$53($]&#51(12!404),12!)3E1/(%1(12`Q
M,C@M,34Y(%!/4TE424].4R`M+2!#3U,H4RD@25,@5$A54R!324XH4RLS,BDN
M#2C)(%1!2T4@041604Y404=%($]&(%1(12!&04-4(%1(050@4TE.*%,I($%.
M1"!#3U,H4RD@05)%(%)%3$%4140@0ED-02!&04-43U(@3T8@4$DO,BX@(-=%
M4D4@R2!334%25"#)(%=/54Q$($A!5D4@5$%+14X@041604Y404=%($]&(%1(
M10U2149,14-424].(%-934U415)9($]&(%-)3B]#3U,L($%.1"!3059%1"!!
M3D]42$52(#8T($)95$53+B`@ST@@5T5,3"XI#0#42$E3(%1(14X@3$5!5D53
M(#DV($)95$53($9/4B!!(%!23TI%0U1)3TX@5$%"3$4L(%=(24-(($E3#4I5
M4U0@5TA!5"#)($Y%140@1D]2(%1(12`TQ"!/0DI%0U0N("#42%53(,D@0T%.
M($U!4T@@5$%"3$53($]&#5@J4TE.*%,I+"!8*D-/4RA3*2P@04Y$($U9(%!2
M3TI%0U1)3TX@5$%"3$4@3T8@1B!8*%HI/40J*%HM6C`I("H@6"!)3E1/#4$@
M4TE.1TQ%(%!!1T4N("#42$E3(%!!1T4@25,@5$A%3B!%6%1%3D1%1"!&4D]-
M("0V,#`P(%1/("3#,#`P+`U)+D4N($=)5DE.1R`Y-B!404),15,L($9/4B!!
M(%1/5$%,($]&(#(T2RX@(,%#0T534TE.1R!42$4@5$%"3$53#4E3($Y/5R!4
M4DE624%,.B!35$]212!8*R0V,"!)3B!42$4@2$E'2"!"651%($]&($$@6D52
M3R!004=%(%!/24Y415(L#51(12!,3U<@0EE412!#3TY404E.4R!42$4@3T9&
M4T54($E.5$\@5$A%(%1!0DQ%("@P($9/4B!42$4@4TE.12!404),12P-,S(@
M1D]2(%1(12!#3U-)3D4@5$%"3$4L($%.1"`Q-C`@1D]2(%1(12!04D]*14-4
M24].(%1!0DQ%*2P@04Y$#41/($%.(,S$P2`HVM`I+-D@5$\@1T54(%1(12!2
M24=(5"!604Q512X-`-1(55,@4D]4051)3TY3($%.1"!04D]*14-424].4R!!
M4D4@3D]7(%9%4ED@1D%35"!!3D0@5D5260U#3TU004-4+B`@SD]412!42$%4
M($E4($E33B=4(%)%04Q,62!.14-%4U-!4ED@5$\@1T5.15)!5$4@02!#3TU0
M3$5410U404),12!/1B!324Y%4R!!3D0@0T]324Y%4RX@(,9/4B!)3E-404Y#
M12P@,3)+($]&(%1!0DQ%4R`H3U(@-DL@3U(-5TA!5$5615(I($-/54Q$($)%
M(%53140L($%.1"!42$4@1DE.04P@4D5354Q4(%-)35!,62!-54Q425!,245$
M($)9#5173RP@3U(@1D]54BX@(,5614X@5$A/54=((%1(12!&24Y!3"!#3T]2
M1$E.051%4R!-24=(5"!204Y'10U&4D]-("TT."XN-#@L($-!3$-53$%424].
M4R!$3TXG5"!.145$(%1/($)%($1/3D4@55-)3D<@5$A%($953$P@4D%.1T4N
M#0#42$4@3$E.12!23U5424Y%($E3(%1(12!'3T]$("=/3"!#2%5.2UD@3$E.
M12!23U5424Y%($923TT-5$A%($Q!4U0@0U5"13-$(%!23T=204TN("#)5"!/
M1B!#3U524T4@2$%$(%1/($)%($U/1$E&245$(%1/(%=/4DL-5TE42"!42$4@
M5%=/($)51D9%4E,@04Y$(%-50T@L($%.1"#)(%)%34]6140@02!"54Y#2"!/
M1B!214%,3%D@4D5$54Y$04Y4#4-/1$4@5$A!5"!705,@24X@5$A%4D4@*-+%

MP<S,V2!214153D1!3E0I+"!%4U!%0TE!3$Q9($E.(%1(12!!0U1504P-1%)!
M5TE.1R!005)4("A-04-23U,@V-/4Q=`@04Y$(-G3U,70("TM($Q)3D53($%2
M12!#3TU-14Y4140@3U54(%=)5$@@02`G*B<I+@W)($%,4T\@041$140@02!$
M3U14140M3$E.12!#05!!0DE,2519("A)5"!/3DQ9(%1!2T53($$@1D57($58
M5%)!#4E.4U1254-424].4RDL(%1/($U!2T4@5$A)3D=3($5!4TE%4B!43R!3
M144N#0#/3DQ9($$@4TE.1TQ%(#/$($]"2D5#5"!)4R!!0U1504Q,62!$4D%7
M3B`M+2!42$4@3U1(15)3#4%212!'14Y%4D%4140@5DE!(%-934U%5%)9("A2
M149,14-424].4R!42%)/54=((%@],"!!3D0@63TP*2X@(--)3D-%(%1(10TS
MQ"!/0DI%0U13($%212!$4D%73B!/3B!!($U50T@@4TU!3$Q%4B!'4DE$+"!4
M2$59($Y%140@5$\@0D4@4T-!3$5$#41/5TX@02!"250N("#)3E-414%$($]&
M(%=2251)3D<@4T5005)!5$4@4D]55$E.15,@5$\@1$5!3"!7251((%1(12`S
MQ`U!3D0@-,0@3T)*14-44RP@R2!324U03%D@4T54(%1(12`TQ"!#3T]21$E.
M051%($]&($5!0T@@4$])3E0@24X@5$A%(#/$#4]"2D5#5"!43R!33TU%($%0
M4%)/4%))051%($Y534)%4BX@(-)%0T%,3"!42$%4($E.($$@,\0@4%)/2D5#
M5$E/3BP-5$A%($9!4E1(15(@05=!62!&4D]-(%E/52!42$4@3T)*14-4($E3
M(%1(12!334%,3$52($E4($=%5%,N("#42$E3#4E3(%1(12!304U%(E14$@
M+2T@5$A%($]"2D5#5"!)4R!055-(140@1$]73B!42$4@-,0@05A)4RP@04Y$
M(%1(25,-2$%3(%1(12!%1D9%0U0@3T8@4TA224Y+24Y'(%1(12!/0DI%0U0@
M55!/3B!04D]*14-424].+@T`V4]5($U!62!(059%($Y/5$E#140@5$A!5"!4
M2$4@,\0@3T)*14-44R!414Y$(%1/($%63TE$(%1(10U#14Y415(@3T8@5$A%
M(%-#4D5%3B`M+2!42$E3($E3($$@0T].4T51545.0T4@3T8@5$A%(%)!3D1/
M32!.54U"15(-1T5.15)!5$]2(,D@0T]$140@55`@*$%.1"!$240@3D]4(%1%
M4U0@1D]2(%-014-44D%,(%!23U!%4E1)15,-3U(@04Y95$A)3D<@3$E+12!4
M2$%4(#HI+B`@SU))1TE.04Q,62#)(%=!4R!'3TE.1R!43R!03$%#12!42$E.
M1U,-24X@02!204Y$3TT@4D]7($%.1"!#3TQ534XL($)55"!42$5.(%1(24Y'
M4R!*55-4($-,54U0140@04Q/3D<@00U$24%'3TY!3"!,24Y%(#HI+B`@R2!7
M24Q,($%,4T\@4T%9(%1(050@5$A%(-/0S,'4(%)/551)3D4@0T%54T5$#4U%
M($U!3ED@1$%94R!/1B!(14%$04-(15,@+2T@5TA/4T4@241%02!705,@250@
M5$\@4%54($-/3$]2($U%34]260U33R!#3$]312!43R!!(,/)P3\@.BD-`,].
M12!42$E.1R#)($A!1"!43R!04E5.12!/550@5T%3($$@4D]55$E.12!72$E#
M2"!$4D%74PU#25)#3$53($%3(%1(12!324Y%+T-/4TE.12!404),15,@05)%
M($)%24Y'(%-%5"!54"X@(,E4($E3#4M)3D1!($Y%050@04Y$($=!5D4@4T]-
M151(24Y'(%1/(%=!5$-((%=(24Q%(%1(12!#3T1%(%=!4PU3151424Y'(%50
M+"!!3D0@04Q33R!705,@02!#2$5#2R!42$%4(%1(12!44DE'(%1!0DQ%4R!7
M15)%($)%24Y'#5-%5"!54"!#3U)214-43%DN("#!3EE705E3+"!!3$P@250@
M1$]%4R!)4R!43R!$4D%7($-/3D-%3E1224,-0TE20TQ%4R!/1B!04D]'4D53
M4TE614Q9($Q!4D=%4B!2041)22P@1D]2($$@4T]25"!/1B!454Y.14Q)4T@M
M3$]/2TE.1PU42$E.1R#)(%-54%!/4T4N#0#42$5212!)4R!!($Q)5%1,12`B
M1D%)3%-!1D4B($E.(%1(12!04D]*14-424].(%)/551)3D4N("#)1@U#3T]2
M1$E.051%4R!!4D4@3U54($]&(%)!3D=%("A'4D5!5$52(%1(04X@.38@3U(@
M-#`I($%&5$52(%!23TI%0U1)3TXL#51(15D@05)%(%-%5"!43R!42$4@3U))
M1TE.+B`@P50@3$5!4U0@3TY%($]&(%1(12!/0DI%0U13(%-#4D574R!54`U&
M4D]-(%1)344@5$\@5$E-12`H5$A%($]#5$%(14123TX@25,@5$A%($U!24X@
M0U5,4%))5"#)(%1(24Y+*2P@04Y$#<D@5$A)3DL@5TA!5"!(05!014Y3($E3
M(%1(050@5$A%($Q)3D4@4D]55$E.12!42$E.2U,@250@3D5%1%,@5$\@1%)!
M5PU!($Q/5"!-3U)%(%!/24Y44R!42$%.($E4(%)%04Q,62!.145$4R!43RX@
M(--/($E4($A!4%!)3%D@34]615,@04Q/3D<-4U1)0TM)3D<@0EE415,@24Y4
M3R!42$4@5%))1R]04D]*14-424].(%1!0DQ%4RP@04Y$($5614X@34%+15,@
M2513#5=!62!54"!43R#6R<,L(-/)Q"!!3D0@5$A%(,/)P5,A("#/3D-%($E4
M($%#5%5!3$Q9(%-405)4140@4$5'1TE.1R!42$4-T\G$(%9/3%5-12!214=)
M4U1%4B!/4B!33TU%5$A)3D<L($)%0T%54T4@5$A%4D4@5T]53$0@0D4@02!0
M15))3T1)0PU,3U5$(%1)0TM)3D<@1E)/32!42$4@4U!%04M%4BX@(,5614Y4
M54%,3%D@5$A%($-/1$4@2E535"!'4DE.1%,@5$\-02!(04Q4($]2($5,4T4@
M0T]-4$Q%5$5,62!(3U-%4R!42$4@4UE35$5-("TM($A%3D-%+"!42$4@1D%)
M3%-!1D4@.BDN#0#&24Y!3$Q9+"!42$4@5D5262!&25)35"!,24Y%4R!/1B!4
M2$4@0T]$12!2141)4D5#5"!42$4-PL'3R<,@5D5#5$]2($%4("0P,S`R+R0P
M,S`S($%.1"#*S=!3(%1/(%1(12#.S<D@TM,OTL73U,_2Q2!23U5424Y%#2A!
M3%1(3U5'2"!!(,+2RR!73U5,1"!04D]"04),62!(059%(%-51D9)0T5$*2X@
M(-1(25,@25,@5$A%($].3%D-5T%9(,D@0T]53$0@1T54(%1(12!#3T1%(%1/
M(%=/4DL@5TE42"!42$4@0U)53D-(15(@+2T@5TE42$]55"!)5"P-5$A%(%!2
M3T=204T@1T]%4R!)3E1/("+)TM$@3$]#2R(N("##4D]34T)/5R!/1B##4D53
M5"!354='15-4140@5$A!5`W!PL-254Y#2$52($1/15,@3D]4(%!55"!!(,/,
MR2!!5"!42$4@14Y$($]&($E44R!#4E5.0TA)3D<@4D]55$E.12P-04Y$(%1(
M050@5$A)4R!#04X@0T%54T4@4%)/0DQ%35,L($U/4U0@3D]404),62!7251(
M(%1(12##R<%3+@T`R50@5$]/2R`Q,"TQ-2!(3U524R!43R!'150@5$A)3D=3
M(%1/($-254Y#2"!!3D0@5T]22R!#3U)214-43%DN#<E.($A)3D1324=(5"#)
M($-!3B!42$E.2R!/1B!!($)53D-(($]&(%1(24Y'4R!42$%4($-/54Q$($A!
M5D4@0D5%3@U%05-)3%D@1$].12!43R!-04M%($E4(%=/4DLL($)55"!!5"!4
M2$4@5$E-12#)(%=!4R!355)%(%)%3$E%5D5$(%=(14X-250@1DE.04Q,62!'
M3U0@1$]73B!43R`T,#DU($)95$53+B`@S4]204PZ(,$@3$E45$Q%(%1(24Y+
M24Y'($5!4DQ9($].#5-!5D53($U!4U-)5D4@5$E-12!!3D0@149&3U)4($1/
M5TX@5$A%(%)/040N#2`-U$A%(,U54TE##2TM+2TM+2TM+2`-`,9)3D%,3%DL
M($$@5T]21"!!0D]55"!42$4@355324,N("#/4DE'24Y!3$Q9(,D@5T%3($=/
M24Y'(%1/#4-/3E-44E5#5"!!(%-%4DE%4R!/1B!#2$]21%,@5TA)0T@@R2!#
M3U5,1"!-3T153$%412!"1517145.($E.#4$@1D%)4DQ9($9,15A)0DQ%(%=!
M62X@(,D@5T%3(%1(14X@1T])3D<@5$\@0E)%04L@55`@5$A%($-(3U)$4PU)
M3B!!($Y)0T4@5T%9($%.1"!-3U9%($)%5%=%14X@5$A%32!204Y$3TU,62X@
M(,)55"!42$5.($E4($]#0U52140-5$\@344@5@A!5"#)($%,4D5!1%D@2TY%
M5R!!(%!)14-%($]&($U54TE#(%=(24-((%=!4R!!(%-%4DE%4R!/1@U"4D]+
M14X@0TA/4D13($%.1"!33U5.1$5$($E.1DE.251%3%D@34]212!#3T],(%1(
M04X@04Y95$A)3D<@R0U705,@1T])3D<@5$\@04-#241%3E1!3$Q9(%=2251%

```
M+"!33R#)(%53140@250@24Y35$5!1"X@(,5614X@0D545$52+`U42$59($%2
M12!&3U52+4Y/5$4@(D-(3U)$4R(L($)23TM%3B!)3E1/($9/55(@1U)/55!3
M($]&($9/55(@3D]415,-14%#2"`M+2!43T\@1T]/1"!43R!005-3(%50+B`@
MSD]415,@05)%($Q/3TM%1"!54"!)3B!!($9215%514Y#60U404),12P@5$A5
M4R!/3B!-62#0P<P@-C0@5$A%($U54TE#($=%5%,@5%)!3E-03U-%1"!43R!!
M($1)1D9%4D5.5`U+15D@*$E.($%$1$E424].(%1/(%!,05E)3D<@050@5$A%
M(%=23TY'(%-0145$(#HI+@T`R2!$3R!.3U0@3D5#15-305))3%D@4D5#3TU-
M14Y$(%5324Y'(%1(12!23U5424Y%($%3($$@34]$14P-1D]2($1/24Y'(,G2
MT2!)3E1%4E)54%13("TM(,D@2$%$($U!3ED@4%)/0DQ%35,@5TE42"`BR=+1
M($Q/0TLB+"!72$5210U!3B#)TM$@25,@0T].5$E.54]54TQ9($Q!5$-(140L
M($%.1"!#3TY315%514Y43%D@25,@0T].4U1!3E1,60U254Y.24Y'(%1(12!2
M3U5424Y%+"!!3D0@R2!35$E,3"!$3R!.3U0@54Y$15)35$%.1"!72$%4($E3
M($A!4%!%3DE.1RP-3D]2($1/(,D@2$%612!!(%-/3%5424].+@T-S45-3U)9
M(,U!4`TM+2TM+2TM+2TM#0`D,,8P,"TD,,;&Q@#35$%25$E.1R!324Y%*U!2
M3TI%0U1)3TX@5$%"3$4-`"0Q,#`P+20R,C4W`,-/1$4-`"0S,#`P+20T,#`P
M`,-(05)!0U1%4B!3150-`"0V,#`P+23#,,;&`--)3D4L($-/4TE.12P@04Y$
M(%!23TI%0U1)3TX@5$%"3$53#0`DPS$P,"TDP\;&Q@#-25-#+B!605))04),
M15,@04Y$(%1!0DQ%4PT-PT].5$5.5%,@3T8@1$E--"Y,3E@-+2TM+2TM+2T-
M`$1)330``--50DU)5%1%1"!%3E1262!&3U(@-$L@1$5-3R!#3TY415-4#0!$
M24TT+E1%6%0`U$A)4R!&24Q%+"!)3B#0Q=33P\G)($9/4DU!5`T`1$E--%)%
M041-12U254Y-12#/0E9)3U53#0!$24TT+DY!3453`,Q)3DM%4B!.04U%($9)
M3$4@5$\@55-%(%=)5$@@S4523$E.(#$R."`-`$U!24XT+E,``,U!24X@0T]$
M12!&3U(@1$E--`T`3T)*14-44RY3`,-/1$4@5$\@1$5&24Y%+U-%5"!54"!/
M0DI%0U13#0!'4D%02$E#4RY3`-9!4DE/55,@1U)!4$A)0U,@4D]55$E.15,@
M*$Q)3D53+"!&24Q,4RP@151#+BD-`$U54TE#+E,``,E.250@04Y$($U!24X@
2R=+1($U54TE#(%)/551)3D4-
`
end

@(A)gfxcode: Binary for 3-D Dungeon Crawls

begin 644 dungeon.sda
M`0@-"`T`GB@R,#8S*0```*D`C2#0J0"-(="I\84OJ0Z%,*4!*?Z%`:D`C98#
MJ?&%+ZD.A3`@M`@LE@,0#"#`_R20,&"B""#)_R"J"K!6+(,#,`H@TO\DD#!*
M3$@((,S_+)8#,!+.E@.I`(7&(.3_R0#P^<D#\"VI""##_ZG^C8(#($(,K6T#
MS74#T!"M;@/-=@/0"*E/(,P)J4LLJ3\@S`E,-`@@S/^I""##_Z4!"0&%`4QT
MI*(OJ0"=7`/*$/JI,(T``:DZC0$!($(,R0+0U.@@0@R=;`/HX`G0]2R6`Q`#
M('<*($(,R1&PN:H8:02%MZEDC6(,J3:--`NI#(UC#*D+C34+H`"I`(6[J0&%
MO"!"#)D"`2#,"<C*T/.I+)D"`2#,"<BM=`.9`@$@S`FI("#,":D(JJ`!A;B&
MNH2Y($(,($(,($(,(-<)K&P#P`70!:G_C7$#P`+P#<`$\`G``=`#($(,&&"@
M`)B9`,R9`,V9`,Z9`,O(T/&JJ0"->P.-?`.-?0.@!2`5#&J(T/EJ:FJ-B@/)
M&;`TR0#P*:@@%0PN?@,N?P,N@`.(T/&H;H`#;G\#;GX#+GL#+GP#+GT#B-#K
M(.X)Z-"RSGH#&&`LE@,P`6!,TO\X8*UL`\D&L/>M=`/)4/`(R5/P!,E5T.A@
MCA0*H`"MB@/,>@/0'YD`RZU[`YD`S*U\`YD`S:U]`YD`SNYZ`Z(!BID`SV#9
M`,N0`\C0U(PL"J#^(#@*B,``T/@@.`JMB@-,^PFY`,N9`<NY`,^9`<^Y`,R9
M`<RY`,V9`<VY`,Z9`<Y@4U1/TE!!0\M3455%1=I#4E5.0\A3455!4\A#4E5.
M0\@@9`X-54XM`*``KFP#RC`+N5<*,`/(T/C(T/*Y5PI(*7\@S`G(:!#S(&0.
M24Y'+BXN`&".^`J,^@H@_PHL@P,P/JZ$`_`&('`+3/,*KFP#\!W@`?`9X`7P
M!.`#T`H@80RP':YL`Y`*((<+L!20`R!"#.```\`?@`O`#(#,+(!X+&*(`H``L
M@P-@KF\#T!:N<`/0#JYQ`]`&J?^-@P-@SG$#SG`#SF\#8$CNA0--A0,8;74#
MC74#D`/N=@-H8$PV"\G^\`%@J4J--`NI"XTU"VAH3,,*C8<#J5J--`NI"XTU
M"TQ%"XQX"Z!PC#0+H`N,-0NLAP.,A`.-AP/.A`/P`M`,H`"I-HTT"ZD+C34+
MK8<#8*D`C8@#C=(+C88#C8L#[HL#C7<#C7@#C7D#C/L+COD+(!4,D`RLA@.M
MBP,9=P.9=P,.BP.0!BZ+`^Z&`^Z(`ZV(`\D8D`0X3/@+H`#9`,OP!)#-L/&^
M`,SL=P/0&;X`S>QX`]`1O@#.['D#T`FY`,\8H@"@`&#(\,V,T@O,>@.0R_#)
M3,T+`0($"!`@0(",/0R-/PRL@0/0!B!"#(V)`ZV)`SD-#-`"&"0X",C`")`"
MH`",@0.@`*D`*&",7PRL@@/`_I`,&)AE+X4OD`+F,*``L2_(C((#H`!@3&0,
M(+P-(*4-(#4-C5X#C6L#C6`#K5T#C5\#J8N-8@RI#(UC#*UK`QA@(#4-.(UA
M^UE`ZU=`XUB`^UF`Y`GK6`#C6L#((P-K5X#C5P#C6D#K5\#C5T#C6H#K64#
MC6$#K68#C6(#K5T#\#"M7`.%%*U=`PGPA14@+@X@C`VE%2D/"="%%2`N#HU<
M`Z45*0\)X(45("X.C5T#T,NI$8UB#*D-C6,,K5P#C6L#C6`#&&`@EPVP`6"M
M7@.-:0.M7P.-:@,@V0VM80.-7@.M8@.-7P-,BPRI`(U<`ZQH`R`5#"Y<`RY=
M`XC0]*U<`]`XK5T#R0'0,6AHJ0"-;P.-<`.-<0.I_XV#`ZUL`\D%T!:@$"`5
M#"YM`RYN`XC0]*`H(!4,B-#Z.&"M7`-,.PZ@`)']YOW0`N;^8*7]R3S0$:7^
MR0/0"SA@J3R%_:D#A?Y@I?W0`L;^QOVB`*']&&"I`J`!C64#C&8#C&<#J?Z@
M`(UC`XQD`ZD)C6@#8*UF`\D0D`%@A16M90.%%*QJ`ZG@(`D.K&D#J=`@"0ZL
M:P.I\"`)#NYE`]`#[F8#8(T1#J45*0\)`(45F*``I@%XA`&1%(8!6&"E_"D/
M"?"%%:7[A12@`*8!>(0!L12&`5A@2*UH`\D,L!^M8P/0`\YD`\YC`ZUC`PUD
M`]`,[F@##F<#K6<#C60#:&!HC7`.:(UQ#B"`#JW___`&(,P)3&P.((`.;'`.
M[G`.T`/N<0Y@`
M`````````````````````````````````````````````````````````````
M`````````````````````````````"`TV0.`$``0!3!$Y%5S7^+!&2P<E'
M`-8?8`1<2)"DC"`,`H((.!,E0PP!$Z(,*1$F28XX`A2<F3!$B8`923)$@X-%
M@(`E.<I$*5&)@(`[>TZL$1`)0QH!=9)42"!@PYX+9P3DHU$B3YXLV#0H0A%%
MP)X22>)Q"9))EH)H6.XIZC"JPH%J%O;DPE%!Z0](,*`(6!1$`(DA.YX(6#/M
M285\5\Y<&'+B$CTF)\9QB:.KX`]R54YC07;J`HD35W;H![+G@*!?!$+D,X]-
MXP\LVF`@$!"F1)0@%:)$\A<L;-"28;26C1A$0`TK.HS!TAA"A\`L04KY+)$@
M3Z0^V`````(#TRB&!@`&`%`'4%E204U)1/XA``!!@`8!"H`\3!BP8?0'&(!2
M@"C]`?0G%*!*`&(16!"+T()0_VIA6$`)'F2<Q`H`J$0@$P"6=%C384N'780%
MCSH=@O``+!T6$Q:\J@2H3)<*!+K\JP2E")\"<`CTE@:`P&KILB0J5*I2K88B
M2K4KQZA71A&=1G6K-V/6LDEK>&(3JX5F`20$L5U5@A:+R()*,&(1]]<FU*]*
```



```
M?7.;CR_\`1!_+6HIJEM"O.:U`'\`:BWJ8=TLXK6\;:UL4S.@B>JFM[K-K0((
M_IJ):MBW&]7MZSB^V@7@=C2K+0#N:&=:UNFNM:]M3<IO4ZI?PQK7N1:-X49/
M:L^-!M:G9_7O-4#JVML&U+G!-.U9P_O0R$;7JG'H;P+>NI<:OS0-(*UO3N=;
MT+K6MWP0K6I-ZSK9QFZWI6TM:6>'NM36KG>M`QT::$<ZUXEF=:L9(^S",+J,
MZJX.>]#=ZGIW\YWJ2`$`%,1#WU@!C#04$0_:XP8P/L<Q=N*-:SSCF\#8QC$1
M`(^7>!,'\(3&.;[)C'^`^8E3]R.-*.+A=%H`AOT$)SG=Z1!O^I.8\(RF.]/)
MSW#:LYS!A*<ZU0#:QWX.D"1^X0Y]Y`<0+EC!)CZ#32WHNU*@^'0"P((<*(`O
M+/,NEI8_L>\4)XR?DHNDY,+L#S#DSVOQE%PV]GU,N>8`/,B+`9C]`;K\%>`W
MF%'VH2NQB]W8^<%/39FS/QS@[Z1%?<%SALK\7>P?:D\VO?CUSR*@OQIJ+6SA
MRX9^,)=.K2`3#_F+<8-K,[_@5:^VBMA:PYK"OY:I3STUD0Y@Y*(0I:A$+=C0
MCU9T8+(!@&R9^/3;C@^#'84U1W!J^8G+&)L<SP+$/X$)3B@P80[[1"4FRGB%
M/>PS$^RD4<LUQ,-;U:.<P'S.\0#=3UT!TI26$P#UD.GQ86+Q6J+I;:@21P4(
MWN"_H2-LGI*NR0PE.PVP(![N--#/,4"."N`SG'%>WLG`Z8/IZJJ[&NYP/_9Q
M4Y/_@A2H`,`%^B$E"7_LXK5&`#]"E*EURA)=;^.NJ1$@SA_$$I;8/].35HD`
M=H0<`^QFTKIS$"_L'X">,XAX/T)Q!%J9BQRIYO?%^Z7Q?ME\]D"@/1)8+^H>
M(JQCU;>]P#K4?5L"`&.]]^C'^M,I]YPZ]^'Y+@+H<WK=X*=@#>I8?V8C:M:H
MET/]*8^%^?.Q5@*`8'VJ31E(,Y#=[%;X)&69"P-T4.H?@3D/I=!Q2EHDB55L
M+3=@[V(N(%R'?:QC,QPTQ>\G/@.=,#E8`"Z'*JXHB7^HH@M?^,*5D]TP+'3]
MX5=7\#,`JED+Z$/UNO@//("#/^B!JA(`J"4"*/@#/S>1EB5R-Z@"/^+),4#`
MOK(!*VA*.O`'HB6Z14`K#JA"G:6"*E1#/_1#G3^`OY9`M3_`OO<:L*;KI('K
MJ);(DB=[OE9HB]:@EB<C'JQKD^[`7W![5_!HKU%[?H`$C]8887`!&*P;68B!
M*J0;M59Z/J1[2W`'KZ'`G_>*M-F;HHE:&*@#)>)!;Q@"L*`!,+S[$FS!*KR'
M21@C'J3".RS#.KR2?)B';9B$`ZA"%72),%<[CCAHLFF:K2/`:NA`&A>#NJ2!
M.NR#!>*!7\0",*C!/HS#.NB#/!='0L0C7B2%%UR+,%=&9B2`6OC;"6=2K^K:
MKS%IDI0*-VCQ1-3PN=B+$/Q%.1!$):A&`U]$/91&K1U%N^Y'.4!P/D]&3S1&
M=_3:3Y387Y0!`#_PA=SP0@RI>LR!5BS$E#%%"QU'FXC$O,K+%S_%L@K&:NQY
M;[1&O_W'KI2*:QS&KK9"*0(````````````````````````````````````````
M`````````````````````````````````````````````````````````````
M```"`7T<2@```0!0"D=204985$%"3$7^(0#^-`-_.S\\_SR_/7\^/S__/[]`
M?T$_0O]"OT-_1#]%_T6_1G]'/TC_2+])?TH_2_]+OTQ_33]._TZ_3W]0/U'^
M#``````````````````````````````````````````````````````````````
M`````````````````````````````````````````````````````````````
M`````````````````````````````````````````````````````````````
M`````````````````````````````````````````(#^F7:!``#`%`*,T1-
M05!35$]26?XA```,`!`0)%C0$,%#A13@#V@PX/X``P`>)%*4"-$BAXT=/42\
M6,'#10L?-4Z$>/Y`ADX8/V[XL%&3)8J;+&76&&$SI$V5*%/";)ES)4F4.EW:
MT/X*)0X9$6R>#-D+Y,Q9*E2A-#621XQ=O43R,'GBUJT3Q5_Y^*4K)DW3L%U)
MW``0(((```B`X*``A@T70'0@"8-U[0I89#@$@F`B@@`?2'R@8,"#!0Y%2"@$T
MH)`@3P*'`",Z#0I0@$39`Q+X82@`H00*'.1<"&``RD0!&"4DH`"A`$4`N!LD
M@@(!W2T(!8@T8"`H5``8/%FP__9V@@``)&<'5O@B```4+?@!X2*-`[($`%1($
M\)#W@`>&`#((\-#7`T?P`7I?S&C[`X$*#CB@P)L+L`.@`^,ZR/SI#R2P-KH#
MJ!Z`ZA^@FP##!@MU8'OD!@B6`P6H0Q`X;R1`X3MKIP&@FF`X,#Z:"SB`P-KC
MD\LZ`QB5R_9$8H)KD$$`NBL@@P(,H""#"3RH\24.C'%@@!`^*-#I"PS(P$/C
M9L)F`).:\R!`$"\(0#KDF3T0A`BDT68E&:7^D40(H-NN`Z=[CG8#[H#=^H<#
M?&XQA`$!<"8#"I=']@1LI]WVV@\!!EXE[)`Y$'A32(3!)^%J`:"X'DS",88`
M"'44`.EM8DF`;I"^QH(">6#6`Y8JH*``G")@@!IO@!WV&.V6Y>G&9KT]=AEE
MO(DFEPF%7="*Y!V0@`,')02@`N0\D,!`UV)L*8281*;``04\F!!"`*W1QH8!
M0%/1QYAC@B"!6GR`(,/5%`7\`>$^2)6!!`A_KED,)7]`0T`!*)X!QU^Z@X`$
M7,*`-094_[Y``73S(#4#!"@4``H%&%!F#D&7@)H&!;#6``<J!6`#(```````
M`````````````````````````````````````````````````````````````
<`````````````````````````````````````
`
end
```

==========

@(#): bottom