The Story of Mel

This was posted to Usenet by its author, Ed Nather (<nather@astro.as.utexas.edu>), on May 21, 1983.

```
A recent article devoted to the macho side of programming
made the bald and unvarnished statement:

    Real Programmers write in FORTRAN.

Maybe they do now,
in this decadent era of
Lite beer, hand calculators, and ``user-friendly'' software
but back in the Good Old Days,
when the term ``software'' sounded funny
and Real Computers were made out of drums and vacuum tubes,
Real Programmers wrote in machine code.
Not FORTRAN.  Not RATFOR.  Not, even, assembly language.
Machine Code.
Raw, unadorned, inscrutable hexadecimal numbers.
Directly.

Lest a whole new generation of programmers
grow up in ignorance of this glorious past,
I feel duty-bound to describe,
as best I can through the generation gap,
how a Real Programmer wrote code.
I'll call him Mel,
because that was his name.

I first met Mel when I went to work for Royal McBee Computer Corp.,
a now-defunct subsidiary of the typewriter company.
The firm manufactured the LGP-30,
a small, cheap (by the standards of the day)
drum-memory computer,
and had just started to manufacture
the RPC-4000, a much-improved,
bigger, better, faster --- drum-memory computer.
Cores cost too much,
and weren't here to stay, anyway.
```

(That's why you haven't heard of the company,
or the computer.)

I had been hired to write a FORTRAN compiler
for this new marvel and Mel was my guide to its wonders.
Mel didn't approve of compilers.

``If a program can't rewrite its own code'',
he asked, ``what good is it?''

Mel had written,
in hexadecimal,
the most popular computer program the company owned.
It ran on the LGP-30
and played blackjack with potential customers
at computer shows.
Its effect was always dramatic.
The LGP-30 booth was packed at every show,
and the IBM salesmen stood around
talking to each other.
Whether or not this actually sold computers
was a question we never discussed.

Mel's job was to re-write
the blackjack program for the RPC-4000.
(Port? What does that mean?)
The new computer had a one-plus-one
addressing scheme,
in which each machine instruction,
in addition to the operation code
and the address of the needed operand,
had a second address that indicated where, on the revolving drum,
the next instruction was located.

In modern parlance,
every single instruction was followed by a GO TO!
Put *that* in Pascal's pipe and smoke it.

Mel loved the RPC-4000
because he could optimize his code:
that is, locate instructions on the drum

so that just as one finished its job,
the next would be just arriving at the ``read head''
and available for immediate execution.
There was a program to do that job,
an ``optimizing assembler'',
but Mel refused to use it.

``You never know where it's going to put things'',
he explained, ``so you'd have to use separate constants''.

It was a long time before I understood that remark.
Since Mel knew the numerical value
of every operation code,
and assigned his own drum addresses,
every instruction he wrote could also be considered
a numerical constant.
He could pick up an earlier ``add'' instruction, say,
and multiply by it,
if it had the right numeric value.
His code was not easy for someone else to modify.

I compared Mel's hand-optimized programs
with the same code massaged by the optimizing assembler program,
and Mel's always ran faster.
That was because the ``top-down'' method of program design
hadn't been invented yet,
and Mel wouldn't have used it anyway.
He wrote the innermost parts of his program loops first,
so they would get first choice
of the optimum address locations on the drum.
The optimizing assembler wasn't smart enough to do it that way.

Mel never wrote time-delay loops, either,
even when the balky Flexowriter
required a delay between output characters to work right.
He just located instructions on the drum
so each successive one was just *past* the read head
when it was needed;
the drum had to execute another complete revolution
to find the next instruction.
He coined an unforgettable term for this procedure.

Although ``optimum'' is an absolute term,
like ``unique'', it became common verbal practice
to make it relative:
``not quite optimum'' or ``less optimum''
or ``not very optimum''.
Mel called the maximum time-delay locations
the ``most pessimum''.

After he finished the blackjack program
and got it to run
(``Even the initializer is optimized'',
he said proudly),
he got a Change Request from the sales department.
The program used an elegant (optimized)
random number generator
to shuffle the ``cards'' and deal from the ``deck'',
and some of the salesmen felt it was too fair,
since sometimes the customers lost.
They wanted Mel to modify the program
so, at the setting of a sense switch on the console,
they could change the odds and let the customer win.

Mel balked.
He felt this was patently dishonest,
which it was,
and that it impinged on his personal integrity as a programmer,
which it did,
so he refused to do it.
The Head Salesman talked to Mel,
as did the Big Boss and, at the boss's urging,
a few Fellow Programmers.
Mel finally gave in and wrote the code,
but he got the test backwards,
and, when the sense switch was turned on,
the program would cheat, winning every time.
Mel was delighted with this,
claiming his subconscious was uncontrollably ethical,
and adamantly refused to fix it.

After Mel had left the company for greener pa$ture$,
the Big Boss asked me to look at the code

and see if I could find the test and reverse it.
Somewhat reluctantly, I agreed to look.
Tracking Mel's code was a real adventure.

I have often felt that programming is an art form,
whose real value can only be appreciated
by another versed in the same arcane art;
there are lovely gems and brilliant coups
hidden from human view and admiration, sometimes forever,
by the very nature of the process.
You can learn a lot about an individual
just by reading through his code,
even in hexadecimal.
Mel was, I think, an unsung genius.

Perhaps my greatest shock came
when I found an innocent loop that had no test in it.
No test. *None*.
Common sense said it had to be a closed loop,
where the program would circle, forever, endlessly.
Program control passed right through it, however,
and safely out the other side.
It took me two weeks to figure it out.

The RPC-4000 computer had a really modern facility
called an index register.
It allowed the programmer to write a program loop
that used an indexed instruction inside;
each time through,
the number in the index register
was added to the address of that instruction,
so it would refer
to the next datum in a series.
He had only to increment the index register
each time through.
Mel never used it.

Instead, he would pull the instruction into a machine register,
add one to its address,
and store it back.
He would then execute the modified instruction

right from the register.
The loop was written so this additional execution time
was taken into account ---
just as this instruction finished,
the next one was right under the drum's read head,
ready to go.
But the loop had no test in it.

The vital clue came when I noticed
the index register bit,
the bit that lay between the address
and the operation code in the instruction word,
was turned on ---
yet Mel never used the index register,
leaving it zero all the time.
When the light went on it nearly blinded me.

He had located the data he was working on
near the top of memory ---
the largest locations the instructions could address ---
so, after the last datum was handled,
incrementing the instruction address
would make it overflow.
The carry would add one to the
operation code, changing it to the next one in the instruction set:
a jump instruction.
Sure enough, the next program instruction was
in address location zero,
and the program went happily on its way.

I haven't kept in touch with Mel,
so I don't know if he ever gave in to the flood of
change that has washed over programming techniques
since those long-gone days.
I like to think he didn't.
In any event,
I was impressed enough that I quit looking for the
offending test,
telling the Big Boss I couldn't find it.
He didn't seem surprised.

```
When I left the company,
the blackjack program would still cheat
if you turned on the right sense switch,
and I think that's how it should be.
I didn't feel comfortable
hacking up the code of a Real Programmer.
```

This is one of hackerdom's great heroic epics, free verse or no. In a few spare images it captures more about the esthetics and psychology of hacking than all the scholarly volumes on the subject put together. For an opposing point of view, see the entry for **Real Programmer**.

[1992 postscript -- the author writes: "The original submission to the net was not in free verse, nor any approximation to it -- it was straight prose style, in non-justified paragraphs. In bouncing around the net it apparently got modified into the `free verse' form now popular. In other words, it got hacked on the net. That seems appropriate, somehow." The author adds that he likes the `free-verse' version better than his prose original...]

[1999 update: Mel's last name is now known. The manual for the LGP-30 refers to "Mel Kaye of Royal McBee who did the bulk of the programming [...] of the ACT 1 system".]

[2001: The Royal McBee LPG-30 turns out to have one other claim to fame. It turns out that meteorologist Edward Lorenz was doing weather simulations on an LGP-30 when, in 1961, he discovered the "Butterfly Theorem" and computational chaos. This seems, somehow, appropriate.]

Real Programmer n.

[indirectly, from the book "Real Men Don't Eat Quiche"] A particular sub-variety of hacker: one possessed of a flippant attitude toward complexity that is arrogant even when justified by experience. The archetypal `Real Programmer' likes to program on the **bare metal** and is very good at same, remembers the binary opcodes for every machine he has ever programmed, thinks that HLLs are sissy, and uses a debugger to edit his code because full-screen editors are for wimps. Real Programmers aren't satisfied with code that hasn't been **bum**med into a state of **tense**ness just short of rupture. Real Programmers never use comments or write documentation: "If it was hard to write", says the Real Programmer, "it should be hard to understand." Real Programmers can make machines do things that were never in their spec sheets; in fact, they are seldom really happy unless doing so. A Real Programmer's code can awe with its fiendish brilliance, even as its crockishness appalls. Real Programmers live on junk food and coffee, hang line-printer art on their walls, and terrify the crap out of other programmers -- because someday, somebody else might have to try to understand their code in order to change it. Their successors generally consider it a **Good Thing** that there aren't many Real Programmers around any more. For a famous (and somewhat more positive) portrait of a Real Programmer, see "**The Story of Mel**" in Appendix A. The term itself was popularized by a letter to the editor in the July 1983 Datamation titled "Real Programmers Don't Use Pascal" by Ed Post, still circulating on Usenet and Internet in on-line form. Typing "Real Programmers Don't Use Pascal" into a web search engine should turn up a copy.

Real Programmers Don't Use Pascal

[A letter to the editor of *Datamation*, volume 29 number 7, July 1983. I've long ago lost my dog-eared photocopy, but I believe this was written (and is copyright) by Ed Post, Tektronix, Wilsonville OR USA.

The story of Mel is a related article.]

Back in the good old days-- the "Golden Era" of computers-- it was easy to separate the men from the boys (sometimes called "Real Men" and "Quiche Eaters" in the literature). During this period, the Real Men were the ones who understood computer programming, and the Quiche Eaters were the ones who didn't. A real computer programmer said things like "DO 10 I=1,10" and "ABEND" (they actually talked in capital letters, you understand), and the rest of the world said things like "computers are too complicated for me" and "I can't relate to computers-- they're so impersonal". (A previous work [1] points out that Real Men don't "relate" to anything, and aren't afraid of being impersonal.)

But, as usual, times change. We are faced today with a world in which little old ladies can get computers in their microwave ovens, 12 year old kids can blow Real Men out of the water playing Asteroids and Pac-Man, and anyone can buy and even understand their very own personal Computer. The Real Programmer is in danger of becoming extinct, of being replaced by high school students with TRASH-80s.

There is a clear need to point out the differences between the typical high school junior Pac-Man player and a Real Programmer. If this difference is made clear, it will give these kids something to aspire to-- a role model, a Father Figure. It will also help explain to the employers of Real Programmers why it would be a mistake to replace the Real Programmers on their staff with 12 year old Pac-Man players (at a considerable salary savings).

The easiest way to tell a Real Programmer from the crowd is by the programming language he (or she) uses. Real Programmers use Fortran. Quiche Eaters use Pascal. Nicklaus Wirth, the designer of Pascal, gave a talk once at which he was asked, "How do you pronounce your name?". He replied, "You can either call me by name, pronouncing it 'Veert', or call me by value, 'Worth'." One can tell immediately by this comment that Nicklaus Wirth is a Quiche Eater. The only parameter passing mechanism endorsed by Real Programmers is call-by-value-return, as implemented in the IBM/370 Fortran G and H compilers. Real Programmers don't need all these abstract concepts to get their jobs done-- they are perfectly happy with a keypunch, a Fortran IV compiler, and a beer.

- Real Programmers do List Processing in Fortran.
- Real Programmers do String Manipulation in Fortran.
- Real Programmers do Accounting (if they do it at all) in Fortran.

- Real Programmers do Artificial Intelligence programs in Fortran.

If you can't do it in Fortran, do it in assembly language. If you can't do it in assembly language, it isn't worth doing.

The academics in computer science have gotten into the "structured programming" rut over the past several years. They claim that programs are more easily understood if the programmer uses some special language constructs and techniques. They don't all agree on exactly which constructs, of course, and the example they use to show their particular point of view invariably fit on a single page of some obscure journal or another-- clearly not enough of an example to convince anyone. When I got out of school, I thought I was the best programmer in the world. I could write an unbeatable tic-tac-toe program, use five different computer languages, and create 1000 line programs that WORKED (Really!). Then I got out into the Real World. My first task in the Real World was to read and understand a 200,000 line Fortran program, then speed it up by a factor of two. Any Real Programmer will tell you that all the Structured Coding in the world won't help you solve a problem like that-- it takes actual talent. Some quick observations on Real Programmers and Structured Programming:

- Real Programmers aren't afraid to use GOTOs.
- Real Programmers can write five page long DO loops without getting confused.
- Real Programmers like Arithmetic IF statements-- they make the code more interesting.
- Real Programmers write self-modifying code, especially if they can save 20 nanoseconds in the middle of a tight loop.
- Real Programmers don't need comments-- the code is obvious.
- Since Fortran doesn't have a structured IF, REPEAT ... UNTIL, or CASE statement, Real Programmers don't have to worry about not using them. Besides, they can be simulated when necessary using assigned GOTOs.

Data structures have also gotten a lot of press lately. Abstract Data Types, Structures, Pointers, Lists, and Strings have become popular in certain circles. Wirth (the above mentioned Quiche Eater) actually wrote an entire book [2] contending that you could write a program based on data structures, instead of the other way around. As all Real Programmers know, the only useful data structure is the Array. Strings, Lists, Structures, Sets-- these are all special cases of arrays and can be treated that way just as easily without messing up your programming language with all sorts of complications. The worst thing about fancy data types is that you have to declare them, and Real Programming Languages, as we all know, have implicit typing based on the first letter of the (six character) variable name.

What kind of operating system is used by a Real Programmer? CP/M? God forbid-- CP/M, after all, is basically a toy operating system. Even little old ladies and grade school students can understand and use CP/M.

Unix is a lot more complicated of course-- the typical Unix hacker never can remember what the PRINT command is called this week-- but when it gets right down to it, Unix is a glorified video game. People don't do Serious Work on Unix systems: they send jokes around the world on UUCP-net and write Adventure games and research papers.

No, your Real Programmer uses OS/370. A good programmer can find and understand the description of the IJK305I error he just got in his JCL manual. A great programmer can write JCL without referring to the manual at all. A truly outstanding programmer can find bugs buried in a 6 megabyte core dump without using a hex calculator. (I have actually seen this done.)

OS is a truly remarkable operating system. It's possible to destroy days of work with a single misplaced space, so alertness in the programming staff is encouraged. The best way to approach the system is through a keypunch. Some people claim there is a Time Sharing system that runs on OS/370, but after careful study I have come to the conclusion that they were mistaken.

What kind of tools does a Real Programmer use? In theory, a Real Programmer could run his programs by keying them into the front panel of the computer. Back in the days when computers had front panels, this was actually done occasionally. Your typical Real Programmer knew the entire bootstrap loader by memory in hex, and toggled it in whenever it got destroyed by his program. (Back then, memory was memory-- it didn't go away when the power went off. Today, memory either forgets things when you don't want it to, or remembers things long after they're better forgotten.) Legend has it that Seymore Cray, inventor of the Cray I supercomputer and most of Control Data's computers, actually toggled the first operating system for the CDC7600 in on the front panel from memory when it was first powered on. Seymore, needless to say, is a Real Programmer.

One of my favorite Real Programmers was a systems programmer for Texas Instruments. One day, he got a long distance call from a user whose system had crashed in the middle of saving some important work. Jim was able to repair the damage over the phone, getting the user to toggle in disk I/O instructions at the front panel, repairing system tables in hex, reading register contents back over the phone. The moral of this story: while a Real Programmer usually includes a keypunch and line printer in his toolkit, he can get along with just a front panel and a telephone in emergencies.

In some companies, text editing no longer consists of ten engineers standing in line to use an 029 keypunch. In fact, the building I work in doesn't contain a single keypunch. The Real Programmer in this situation has to do his work with a "text editor" program. Most systems supply several text editors to select from, and the Real Programmer must be careful to pick one that reflects his personal style. Many people believe that the best text editors in the world were written at Xerox Palo Alto Research Center for use on their Alto and Dorado computers[3]. Unfortunately, no Real Programmer would ever use a computer whose operating system is called SmallTalk, and would certainly not talk to the computer with a mouse.

Some of the concepts in these Xerox editors have been incorporated into editors running on more reasonably named operating systems-- EMACS and VI being two. The problem with these editors is that Real Programmers consider "what you see is what you get" to be just as bad a concept in Text Editors as it is in Women. No, the Real Programmer wants a "you asked for it, you got it" text editor-- complicated, cryptic, powerful, unforgiving, dangerous. TECO, to be precise.

It has been observed that a TECO command sequence more closely resembles transmission line noise than readable text[4]. One of the more entertaining games to play with TECO is to type your name in as a command line and try to guess what it does. Just about any possible typing error while talking with TECO will probably destroy your program, or even worse-- introduce subtle and mysterious bugs in a once working subroutine.

For this reason, Real Programmers are reluctant to actually edit a program that is close to working. They find it much easier to just patch the binary object code directly, using a wonderful program called SUPERZAP (or its equivalent on non-IBM machines). This works so well that many working programs on IBM systems bear no relation to the original Fortran code. In many cases, the original source code is no longer available. When it comes time to fix a program like this, no manager would even think of sending anything less than a Real Programmer to do the job-- no Quiche Eating structured programmer would even know where to start. This is called "job security".

Some programming tools NOT used by Real Programmers:

- Fortran preprocessors like MORTRAN and RATFOR. The Cuisinarts of programming-- great for making Quiche. See comments above on structured programming.
- Source language debuggers. Real Programmers can read core dumps.
- Compilers with array bounds checking. They stifle creativity, destroy most of the interesting uses for EQUIVALENCE, and make it impossible to modify the operating system code with negative subscripts. Worst of all, bounds checking is inefficient.
- Source code maintenance systems. A Real Programmer keeps his code locked up in a card file, because it implies that its owner cannot leave his important programs unguarded [5].

Where does the typical Real Programmer work? What kind of programs are worthy of the efforts of so talented an individual? You can be sure that no Real Programmer would be caught dead writing accounts-receivable programs in COBOL, or sorting mailing lists for People magazine. A Real Programmer wants tasks of earth-shaking importance (literally!).

- Real Programmers work for Los Alamos National Laboratory, writing atomic bomb simulations to run on Cray I supercomputers.
- Real Programmers work for the National Security Agency, decoding Russian transmissions.

- It was largely due to the efforts of thousands of Real Programmers working for NASA that our boys got to the moon and back before the Russkies.
- The computers in the Space Shuttle were programmed by Real Programmers.
- Real Programmers are at work for Boeing designing the operation systems for cruise missiles.

Some of the most awesome Real Programmers of all work at the Jet Propulsion Laboratory in California. Many of them know the entire operating system of the Pioneer and Voyager spacecraft by heart. With a combination of large ground-based Fortran programs and small spacecraft-based assembly language programs, they are able to do incredible feats of navigation and improvisation-- hitting ten-kilometer wide windows at Saturn after six years in space, repairing or bypassing damaged sensor platforms, radios, and batteries. Allegedly, one Real Programmer managed to tuck a pattern matching program into a few hundred bytes of unused memory in a Voyager spacecraft that searched for, located, and photographed a new moon of Jupiter.

The current plan for the Galileo spacecraft is to use a gravity assist trajectory past Mars on the way to Jupiter. This trajectory passes within 80 +/- 3 kilometers of the surface of Mars. Nobody is going to trust a Pascal program (or Pascal programmer) for navigation to these tolerances.

As you can tell, many of the world's Real Programmers work for the U.S. Government-- mainly the Defense Department. This is as it should be. Recently, however, a black cloud has formed on the Real Programmer horizon. It seems that some highly placed Quiche Eaters at the Defense Department decided that all Defense programs should be written in some grand unified language called "ADA" ((C), DoD). For a while, it seemed that ADA was destined to become a language that went against all the precepts of Real Programming-- a language with structure, a language with data types, strong typing, and semicolons. In short, a language designed to cripple the creativity of the typical Real Programmer. Fortunately, the language adopted by DoD had enough interesting features to make it approachable-- it's incredibly complex, includes methods for messing with the operating system and rearranging memory, and Edsger Dijkstra doesn't like it [6]. (Dijkstra, as I'm sure you know, was the author of "GOTOs Considered Harmful"-- a landmark work in programming methodology, applauded by Pascal Programmers and Quiche Eaters alike.) Besides, the determined Real Programmer can write Fortran programs in any language.

The Real Programmer might compromise his principles and work on something slightly more trivial than the destruction of life as we know it. Providing there's enough money in it. There are several Real Programmers building video games at Atari, for example. (But not playing them-- a Real Programmer knows how to beat the machine every time: no challenge in that.) Everyone working at LucasFilm is a Real Programmer. (It would be crazy to turn down the money of fifty million Star Trek fans.) The proportion of Real Programmers in Computer Graphics is somewhat lower than the norm, mostly because nobody has found a use for Computer Graphics

yet. On the other hand, all Computer Graphics is done in Fortran, so there are a fair number of people doing Graphics in order to avoid having to write COBOL programs.

Generally, the Real Programmer plays the same way he works-- with computers. He is constantly amazed that his employer actually pays him to do what he would be doing for fun anyway (although he is careful not to express this opinion out loud). Occasionally, the Real Programmer does step out of the office for a breath of fresh air and a beer or two. Some tips on recognizing Real Programmers away from the computer room:

- At a party, the Real Programmers are the ones in the corner talking about operating system security and how to get around it.
- At a football game, the Real Programmer is the one comparing the plays against his simulations printed on 11 by 14 fanfold paper.
- At the beach, the Real Programmer is the one drawing flowcharts in the sand.
- At a funeral, the Real Programmer is the one saying "Poor George. And he almost had the sort routine working before the coronary."
- In a grocery store, the Real Programmer is the one who insists on running the cans past the laser checkout scanner himself, because he never could trust keypunch operators to get it right the first time.

What sort of environment does the Real Programmer function best in? This is an important question for the managers of Real Programmers. Considering the amount of money it costs to keep one on the staff, it's best to put him (or her) in an environment where he can get his work done.

The typical Real Programmer lives in front of a computer terminal. Surrounding this terminal are:

- Listings of all programs the Real Programmer has ever worked on, piled in roughly chronological order on every flat surface in the office.
- Some half-dozen or so partly filled cups of cold coffee. Occasionally, there will be cigarette butts floating in the coffee. In some cases, the cups will contain Orange Crush.
- Unless he is very good, there will be copies of the OS JCL manual and the Principles of Operation open to some particularly interesting pages.
- Taped to the wall is a line-printer Snoopy calendar for the year 1969.
- Strewn about the floor are several wrappers for peanut butter filled cheese bars-- the type that are made pre-stale at the bakery so they can't get any worse while waiting in the vending machine.
- Hiding in the top left-hand drawer of the desk is a stash of double-stuff Oreos for special occasions.
- Underneath the Oreos is a flow-charting template, left there by the previous occupant of the office. (Real Programmers write programs, not documentation. Leave that to the maintenence people.)

The Real Programmer is capable of working 30, 40, even 50 hours at a stretch, under intense pressure. In fact, he prefers it that way. Bad response time doesn't bother the Real Programmer-- it gives him a chance to catch a little sleep between compiles. If there is not enough schedule pressure on the Real Programmer, he tends to make things more challenging by working on some small but interesting part of the problem for the first nine weeks, then finishing the rest in the last week, in two or three 50-hour marathons. This not only impresses the hell out of his manager, who was despairing of ever getting the project done on time, but creates a convenient excuse for not doing the documentation. In general:

- No Real Programmer works 9 to 5. (Unless it's the ones at night.)
- Real Programmers don't wear neckties.
- Real Programmers don't wear high heeled shoes.
- Real Programmers arrive at work in time for lunch.
- A Real Programmer might or might not know his wife's name. He does, however, know the entire ASCII (or EBCDIC) code table.
- Real Programmers don't know how to cook. Grocery stores aren't open at three in the morning. Real Programmers survive on Twinkies and coffee.

What of the future? It is a matter of some concern to Real Programmers that the latest generation of computer programmers are not being brought up with the same outlook on life as their elders. Many of them have never seen a computer with a front panel. Hardly anyone graduating from school these days can do hex arithmetic without a calculator. College graduates these days are soft-- protected from the realities of programming by source level debuggers, text editors that count parentheses, and "user friendly" operating systems. Worst of all, some of these alleged "computer scientists" manage to get degrees without ever learning Fortran! Are we destined to become an industry of Unix hackers and Pascal programmers?

From my experience, I can only report that the future is bright for Real Programmers everywhere. Neither OS/370 nor Fortran show any signs of dying out, despite all the efforts of Pas- cal programmers the world over. Even more subtle tricks, like adding structured coding constructs to Fortran, have failed. Oh sure, some computer vendors have come out with Fortran 77 compilers, but every one of them has a way of converting itself back into a Fortran 66 compiler at the drop of an option card-- to compile DO loops like God meant them to be.

Even Unix might not be as bad on Real Programmers as it once was. The latest release of Unix has the potential of an operating system worthy of any Real Programmer-- two different and subtly incompatible user interfaces, an arcane and complicated teletype driver, virtual memory. If you ignore the fact that it's "structured", even 'C' programming can be appreciated by the Real Programmer: after all, there's no type checking, variable names are seven (ten? eight?) characters long, and the added bonus of the Pointer data type is thrown in-- like having the best parts of Fortran and assembly language in one place. (Not to mention some of the more creative uses for #define.)

No, the future isn't all that bad. Why, in the past few years, the popular press has even commented on the bright new crop of computer nerds and hackers ([7] and [8]) leaving places like Stanford and MIT for the Real World. From all evidence, the spirit of Real Programming lives on in these young men and women. As long as there are ill-defined goals, bizarre bugs, and unrealistic schedules, there will be Real Programmers willing to jump in and Solve The Problem, saving the documentation for later. Long live Fortran!

References:

[1] Feirstein, B., "Real Men don't Eat Quiche", New York, Pocket Books, 1982.

[2] Wirth, N., "Algorithms + Data Structures = Programs", Prentice Hall, 1976.

[3] Ilson, R., "Recent Research in Text Processing", IEEE Trans. Prof. Commun., Vol. PC-23, No. 4, Dec. 4, 1980.

[4] Finseth, C., "Theory and Practice of Text Editors - or - a Cookbook for an EMACS", B.S. Thesis, MIT/LCS/TM-165, Massachusetts Institute of Technology, May 1980.

[5] Weinberg, G., "The Psychology of Computer Programming", New York, Van Nostrand Reinhold, 1971, p. 110.

[6] Dijkstra, E., "On the GREEN language submitted to the DoD", Sigplan notices, Vol. 3, No. 10, Oct 1978.

[7] Rose, Frank, "Joy of Hacking", Science 82, Vol. 3, No. 9, Nov 82, pp. 58-66.

[8] "The Hacker Papers", Psychology Today, August 1980.

ACKNOWLEGEMENT

I would like to thank Jan E., Dave S., Rich G., Rich E. for their help in characterizing the Real Programmer, Heather B. for the illustration, Kathy E. for putting up with it, and atd!avsdS:mark for the initial inspiration.

SnoopyCalendar

The follow example of a Snoopy Calendar is typical of FORTRAN IV computer programs of the 1960's and 1970's. Various Hacker tests (Joke sites) and the classic "Real Programmers Don't Use Pascal, They Use FORTRAN" article refer to a Snoopy Calendar from 1969 being pinned up on the wall of a 'Real Programmer'. For nostalgic reasons this card deck can be run on the [IBM 1130] emulator on a PC.

The sample output as generated by the IBM 1130 emulator at the bottom of this page shows the arch typical calendar from 1969. Quote from the Real Programmers Don't Use Pascal arcticle "Taped to the wall is a line-printer Snoopy calendar for the year 1969". [A copy of the article can be found here]

This particular piece of computer code has great (computer related) historical significance. The Snoopy Calendar was one of the most widely shared and implemented computer artifacts in the1960's and 1970's.

IBM 1130 Card Deck

Please note: for formating reasons there is an extra space in column 1 which needs to be removed to complile this deck on the IBM 1130 Emulator.

```
 // JOB
 // FOR
 *LIST SOURCE PROGRAM
 *ONE WORD INTEGERS
 *EXTENDED PRECISION
       SUBROUTINE SNPIC
 C     THIS SUBROUTINE WILL ANALYZE THE INPUT DATA AND PRINT A PICTURE
       DIMENSION ILINE(133),INUM(50),ICHR(50)
       COMMON ISET
       DATA IBLNK/' '/
       DO 4 I=1,121
     4 ILINE(I)=IBLNK
       K=1
    10 READ (2,1000) (INUM(I),ICHR(I),I=1,ISET)
       DO 40 I = 1, ISET
       IF (INUM(I)+1)100,11,100
 C     HERE WE WRITE A LINE TO THE PRINTER AND GO BUILD ANOTHER
    11 DO 15 L = K, 121
```

```
   15 ILINE(L)=ICHR(I)
      WRITE (3,2000) (ILINE(K),K=1,121)
      ILINE(1)=IBLNK
      DO 20 K = 2, 121
   20 ILINE(K)=ICHR(I)
      K=1
      I=I+1
  100 ITST=INUM(I)
      IF (ITST + 2) 101,200,101
  101 IF (ITST) 102,40,102
  102 DO 30 J = 1, ITST
      ILINE(K)=ICHR(I)
      K=K+1
   30 CONTINUE
   40 CONTINUE
      GOTO 10
C     HERE WE EXIT THE PICTURE AND RETURN TO THE CALLING PROGRAM
  200 RETURN
C     FORMAT STATEMENTS
 1000 FORMAT (25(I2,A1))
 2000 FORMAT (121A1)
      END
// DUP
*DELETE             SNPIC
*STORE      WS  UA  SNPIC
// JOB
// FOR
*LIST SOURCE PROGRAM
*IOCS(CARD,1132 PRINTER)
*ONE WORD INTEGERS
*EXTENDED PRECISION
C     ***************************************************************
C     * IBM 1130 EMULATOR VERSION (KYM FARNIK)                      *
C     *                                                             *
C     * COPYRIGHT - THIS IS CONSIDERED PUBLIC DOMAIN FOR THE        *
C     * FOLLOWING REASON                                            *
C     *    DURING THE 60'S AND 70'S THIS CODE (AND VARIATIONS)      *
C     *    WAS WIDELY SHARED BETWEEN GROUPS SUCH AS SHARE/GUIDE(IBM) *
C     *    AND DECUS(DIGITAL) AND THE ORGINAL AUTHOR IS UNKNOWN AND  *
C     *    HAS NEVER ASSERTED ANY RIGHTS OVER THIS CODE.            *
C     *                                                             *
```

```
C     * THIS VERSION WAS MODIFIED FROM CODE SOURCED FROM DECUS       *
C     * SEE: http://www.ibiblio.org/pub/academic/computer-science    *
C     *          /history/pdp-11/rsts/decus/sig87/087018/            *
C     * WHICH IN TURN WAS SOURCED FROM AN IBM FORTRAN+BAL VERSION    *
C     *                                                              *
C     * IBM 1130 MODS: (FORTRAN IV SUBSET)                           *
C     *   ADDED IBM 1130 JCL                                         *
C     *   REMOVE OPEN/CLOSE                                          *
C     *   CHANGED LOGIGAL IF TO ARITHMETIC IF                        *
C     *   CHANGED TO 5 CHARACTER VARIABLE NAMES                      *
C     *   CHANGED TO 120 COLUMNS IBM 1132 PRINTER                    *
C     *   EXTENED PRECISION REAL (3*16 BIT WORDS)                    *
C     *                                                              *
C     ****************************************************************
C     * PRINTS CALENDAR, ONE MONTH PER PAGE WITH PICTURES OPTIONAL.  *
C     *                                                              *
C     * BEGINNING MONTH AND YEAR, ENDING MONTH AND YEAR MUST BE PRO- *
C     * VIDED IN 4(I6)  FORMAT ON A CARD IMMEDIATELY FOLLOWING       *
C     * CARD 98 OF DECK.                                             *
C     *                                                              *
C     * IF GRID LINES ARE DESIRED, A 1 MUST APPEAR IN COLUMN 30 OF   *
C     * ABOVE CARD.  A BLANK OR ZERO WILL SUPPRESS GRID LINES.       *
C     *                                                              *
C     * ALL PICTURE DATA DECKS MUST BE TERMINATED WITH CODE -2.      *
C     * CONSECUTIVE -2*S WILL RESULT IN NO PICTURE BEING PRINTED     *
C     * FOR THAT MONTH.                                              *
C     *                                                              *
C     * PICTURE FORMAT CODES --                                      *
C     *    -1    END OF LINE                                         *
C     *    -2    END OF PICTURE                                      *
C     *    -3    LIST CARDS, ONE PER LINE, FORMAT 13A6               *
C     *    -4    LIST CARDS, TWO PER LINE, FORMAT 11A6/11A6          *
C     *    -5    LIST CARDS, TWO PER LINE, FORMAT 12A6/10A6          *
C     ****************************************************************
      DIMENSION AMNTH (12,7,13), ANAM(22), ANUM(2,10,5), NODS(12),
     1 CAL(60,22)
      COMMON ISET
      READ (2,1) (((AMNTH(I,J,K),K=1,13),J=1,7),I=1,12)
      READ (2,2) (ANAM(I),I=1,22)
      READ (2,3) (((ANUM(I,J,K),J=1,10),K=1,5),I=1,2)
      READ (2,4) (NODS(I),I=1,12)
```

```
      READ (2,1) BLANK,ONE,ALIN1,ALIN2,ALIN3,ALIN4
      READ (2,4) MF,IYR,MTLST,IYLST,LNSW
      ISET=25
      DO 10 I=1,60
      DO 10 J=1,22
10    CAL(I,J)= BLANK
      CAL(1,1)= ONE
      DO 20 J=1,22
20    CAL(11,J)=ANAM(J)
      IF (LNSW) 122,142,122
122   DO 125 I=20,60,8
      DO 125 J=1,22
125   CAL(I,J)=ALIN2
      DO 140 J=4,19,3
      I=13
127   DO 130 L=1,7
      CAL(I,J)=ALIN1
130   I=I+1
      IF (I-55) 135,135,140
135   CAL(I,J)=ALIN3
      I=I+1
      GO TO 127
140   CONTINUE
      DO 141 I=20,60,8
141   CAL(I,1)=ALIN4
142   IDOW=(IYR-1751)+(IYR-1753)/4-(IYR-1701)/100+(IYR-1601)/400
      IDOW=IDOW-7*((IDOW-1)/7)
55    IF (IYR-IYLST) 60,65,100
60    ML=12
      GO TO 70
65    ML=MTLST
70    IY1=IYR/1000
      NUMB=IYR-1000*IY1
      IY2=NUMB/100
      NUMB=NUMB-100*IY2
      IY3=NUMB/10
      NUMB=NUMB-10*IY3
      IY4=NUMB
      DO 72 J=1,5
      CAL(J+3,2)=ANUM(2,IY1+1,J)
      CAL(J+1,3)=ANUM(2,IY2+1,J)
```

```
      CAL(J+1,19)=ANUM(2,IY3+1,J)
72    CAL(J+3,20)=ANUM(2,IY4+1,J)
      LPYSW=0
      IF (IYR-4*(IYR/4)) 90,75,90
75    IF (IYR-100*(IYR/100)) 85,80,85
80    IF (IYR-400*(IYR/400)) 90,85,90
85    LPYSW=1
90    NODS(2)=NODS(2)+LPYSW
      IF (MF-1) 100,110,95
95    MF=MF-1
      DO 105 MONTH=1,MF
105   IDOW=IDOW+NODS(MONTH)
      IDOW=IDOW-7*((IDOW-1)/7)
      MF=MF+1
110   DO 51 MONTH=MF,ML
      LSTDY=NODS(MONTH)
      DO 115 I=1,7
      DO 115 JM=1,13
      J=JM+4
115   CAL(I,J)=AMNTH(MONTH,I,JM)
      IF (IDOW-1) 160,160,120
120   ID=IDOW-1
      J=2
      DO 155 K=1,ID
      DO 150 I=14,18
      CAL (I,J)= BLANK
150   CAL(I,J+1)= BLANK
      J=J+3
155   CONTINUE
160   IDAY=1
      II=14
25    J=3*IDOW-1
      N=IDAY/10+1
      I=II
      DO 30 K=1,5
      CAL(I,J)=ANUM(1,N,K)
30    I=I+1
      N=IDAY-10*N+11
      J=J+1
      I=II
      DO 35 K=1,5
```

```
      CAL(I,J)=ANUM(2,N,K)
35    I=I+1
      IDOW=IDOW+1
      IF (IDOW-7) 45,45,40
40    IDOW=1
      II=II+8
45    IDAY=IDAY+1
      IF (IDAY-LSTDY) 25,25,50
50    ID=IDOW
205   I=II
      J=3*ID-1
      DO 210 K=1,5
      CAL(I,J)= BLANK
      CAL(I,J+1)= BLANK
210   I=I+1
      IF (ID-7) 215,220,220
215   ID=ID+1
      GO TO 205
220   IF (II-54) 225,230,230
225   II=54
      ID=1
      GO TO 205
230   CALL SNPIC
      WRITE (3,5) ((CAL(I,J),J=1,21),I=1,60)
51    CONTINUE
      IF (IYR-IYLST) 235,100,100
235   NODS(2)=NODS(2)-LPYSW
      IYR=IYR+1
      MF=1
      GO TO 55
100   CALL EXIT
1     FORMAT (13A6)
2     FORMAT (11A6)
3     FORMAT (10A6)
4     FORMAT (12I6)
5     FORMAT (A1,20A6)
      END
// XEQ
         JJJJJJJ   AAAAA   N     N  U     U   AAAAA   RRRRRR   Y     Y
            J     A     A  NN    N  U     U  A     A  R     R   Y   Y
            J     A     A  N N   N  U     U  A     A  R     R    Y Y
            J     AAAAAAA  N  N  N  U     U  AAAAAAA  RRRRRR      Y
```

```
       J     A     A  N   N N  U     U  A     A  R   R       Y
    J  J     A     A  N    NN  U     U  A     A  R    R      Y
     JJ      A     A  N     N   UUUUU   A     A  R     R     Y
FFFFFFF  EEEEEEE  BBBBBB   RRRRRR   U     U   AAAAA   RRRRRR   Y     Y
F        E        B     B  R     R  U     U  A     A  R     R   Y   Y
F        E        B     B  R     R  U     U  A     A  R     R    Y Y
FFFFF    EEEEE    BBBBBB   RRRRRR   U     U  AAAAAAA  RRRRRR      Y
F        E        B     B  R   R    U     U  A     A  R   R       Y
F        E        B     B  R    R   U     U  A     A  R    R      Y
F        EEEEEEE  BBBBBB   R     R   UUUUU   A     A  R     R     Y
            M     M   AAAAA   RRRRRR    CCCCC   H     H
            MM   MM  A     A  R     R  C     C  H     H
            M M M M  A     A  R     R  C        H     H
            M  M  M  AAAAAAA  RRRRRR   C        HHHHHHH
            M     M  A     A  R   R    C        H     H
            M     M  A     A  R    R   C     C  H     H
            M     M  A     A  R     R   CCCCC   H     H
             AAAAA   PPPPPP   RRRRRR   IIIIIII  L
            A     A  P     P  R     R     I     L
            A     A  P     P  R     R     I     L
            AAAAAAA  PPPPPP   RRRRRR      I     L
            A     A  P        R   R       I     L
            A     A  P        R    R      I     L
            A     A  P        R     R  IIIIIII  LLLLLLL
                      M     M   AAAAA   Y     Y
                      MM   MM  A     A   Y   Y
                      M M M M  A     A    Y Y
                      M  M  M  AAAAAAA     Y
                      M     M  A     A     Y
                      M     M  A     A     Y
                      M     M  A     A     Y
                JJJJJJJ  U     U  N     N  EEEEEEE
                   J     U     U  NN    N  E
                   J     U     U  N N   N  E
                   J     U     U  N  N  N  EEEEE
                   J     U     U  N   N N  E
                J  J     U     U  N    NN  E
                 JJ       UUUUU   N     N  EEEEEEE
                JJJJJJJ  U     U  L        Y     Y
                   J     U     U  L         Y   Y
                   J     U     U  L          Y Y
                   J     U     U  L           Y
                   J     U     U  L           Y
                J  J     U     U  L           Y
                 JJ       UUUUU   LLLLLLL     Y
         AAAAA   U     U   GGGGG   U     U   SSSSS   TTTTTTT
        A     A  U     U  G     G  U     U  S     S     T
        A     A  U     U  G        U     U  S           T
        AAAAAAA  U     U  G   GGG  U     U   SSSSS      T
        A     A  U     U  G     G  U     U        S     T
        A     A  U     U  G     G  U     U  S     S     T
```

```
                A     A   UUUUU    GGGGG    UUUUU    SSSSS      T
 SSSSS   EEEEEEE  PPPPPP  TTTTTTT  EEEEEEE  M     M  BBBBBB   EEEEEEE  RRRRRR
S     S  E        P     P    T     E        MM   MM  B     B  E        R     R
S        E        P     P    T     E        M M M M  B     B  E        R     R
 SSSSS   EEEEE    PPPPPP     T     EEEEE    M  M  M  BBBBBB   EEEEE    RRRRRR
      S  E        P          T     E        M     M  B     B  E        R   R
S     S  E        P          T     E        M     M  B     B  E        R    R
 SSSSS   EEEEEEE  P          T     EEEEEEE  M     M  BBBBBB   EEEEEEE  R     R
            OOOOO    CCCCC   TTTTTTT   OOOOO   BBBBBB   EEEEEEE  RRRRRR
           O     O  C     C     T     O     O  B     B  E        R     R
           O     O  C           T     O     O  B     B  E        R     R
           O     O  C           T     O     O  BBBBBB   EEEEE    RRRRRR
           O     O  C           T     O     O  B     B  E        R   R
           O     O  C     C     T     O     O  B     B  E        R    R
            OOOOO    CCCCC      T      OOOOO   BBBBBB   EEEEEEE  R     R
       N     N   OOOOO   V     V  EEEEEEE  M     M  BBBBBB   EEEEEEE  RRRRRR
       NN    N  O     O  V     V  E        MM   MM  B     B  E        R     R
       N N   N  O     O  V     V  E        M M M M  B     B  E        R     R
       N  N  N  O     O  V     V  EEEEE    M  M  M  BBBBBB   EEEEE    RRRRRR
       N   N N  O     O   V   V   E        M     M  B     B  E        R   R
       N    NN  O     O    V V    E        M     M  B     B  E        R    R
       N     N   OOOOO      V     EEEEEEE  M     M  BBBBBB   EEEEEEE  R     R
       DDDDDD   EEEEEEE   CCCCC   EEEEEEE  M     M  BBBBBB   EEEEEEE  RRRRRR
       D     D  E        C     C  E        MM   MM  B     B  E        R     R
       D     D  E        C        E        M M M M  B     B  E        R     R
       D     D  EEEEE    C        EEEEE    M  M  M  BBBBBB   EEEEE    RRRRRR
       D     D  E        C        E        M     M  B     B  E        R   R
       D     D  E        C     C  E        M     M  B     B  E        R    R
       DDDDDD   EEEEEEE   CCCCC   EEEEEEE  M     M  BBBBBB   EEEEEEE  R     R
         SUNDAY            MONDAY            TUESDAY          WEDN
ESDAY         THURSDAY           FRIDAY           SATURDAY
        1    222  33333    4  55555  666  77777  888   999
       11   2   2     3   44  5     6         7 8   8 9   9
        1      2    33   4 4  5555  6666     7   888   9999
        1    2    3   3 44444     5 6   6   7   8   8     9
      11111 22222  333     4  5555   666    7    888   999
  000    1    222  33333    4  55555  666  77777  888   999
 0   0  11   2   2     3   44  5     6         7 8   8 9   9
 0   0   1      2    33   4 4  5555  6666     7   888   9999
 0   0   1    2    3   3 44444     5 6   6   7   8   8     9
  000  11111 22222  333     4  5555   666    7    888   999
    31    28    31    30    31    30    31    31    30    31    30    31
      1        I  ---------I-- -----
    12  1969    12  1969     1
 11-1                                                                  *PAGE
 1152  1X 1  1X 1  9X-1 51  5X10  20-1 50  4X 1  1X 4  2/ 6  20-1 49  2X 1 SNP 1
 1X 5  2/ 9  20-1 48  4X 5  2/11  20-1 48  2X 6  2/12  20-1 47  2$22  20-1 SNP 2
46  3$23  20-1 45  3$ 1  1$ 9  4* 3  4* 2  20-1 45  3$ 1  1$ 9  5* 2  5* 2 SNP 3
 10-1 45  3$ 1  1$ 9  5* 2  5* 2  20-1 44  6$ 9  5* 2  5* 3  50-1 44  3$ 1 SNP 4
 2$ 9  4* 4  4* 8  30-1 43  4$ 1  2$33  20-1 43  7$35  20-1 43  6$37  30-1 SNP 5
43  7$37  30-1 43  7$38  30-1 42  6$ 1  2$38  30-1 42  7$ 1  2$22  7M 8  30SNP 6
```

```
-1 42  7$ 1  1$ 1  2O19 11M 5  3O-1 42  7$ 1  1$ 2  2O18 11M 5  3O-1 41  8$SNP 7
 1  1$ 3  1O 1  6O14  7M 6  3O-1 41  1$ 1  6$ 1  1$ 4  1O 5  6O21  3O-1 41 SNP 8
 1$ 1  6$ 1  1$ 4  2O 7  3O20  3O-1 41  7$ 2  1$ 5  1O 7  1O 2  3O14  4O-1 SNP 9
41  1$ 4  2$ 2  1$ 6  1O 6  1O 4 16O-1 41  2$ 3  1$ 1  1$ 8  1O 6  1O 4  1$SNP10
 1  1$ 2  2$-1 42  3$ 1  1$ 1  1$ 7 10* 3  1$ 1  1$ 1  3$-1 44  2$11  1O 1 SNP11
 5* 1  2O 2  1$ 2  3$-1 44  2$10  2O 7  2O 2  2$ 1  1$ 1  1$-1 55  2O 8  2OSNP12
 3  2$ 1  1$-1 56  2O 8  2O-1 55  2O10  2O-1 55  2O11  2O-1 54  2O12  2O-1 SNP13
53  2O13  2O-1 51  3O 6  1O 7  2O-1 50  3O 7  1O 7  3O-1 49  3O 2  1O 5  1OSNP14
 8  2O-1 48  3O 2  1O 5  2O 8  2O-1 47  3O 3  1O 5  2O 8  2O-1 46  2O 1  3OSNP15
 1  1O 5  2O 8  2O-1 45  4O 2  3O 5  2O 8  2O-1 44  3O 5  2O 5  2O 7  2O-1 SNP16
44  3O 5  2O 5  2O 7  3O-1 43  2O 7  1O 6  3O 5  2O 1  4O-1 43  2O 7  1O 4 SNP17
 8O 1 14O-1 38  6I 3O 5  1O 7  1O 3  7O 9  2O-1 34  4I 1  6I 2O 5  1O17  3OSNP18
 8  2O-1 32  3I11  7O12  4O 4  2O 3  1O 3  2O10  6--1 32  7I 3  2I 8  1O15 SNP19
 3O 3  2O 3  2O 2  1O 3  8--1 37  3I 3  3I 6  1O10  1O 6  2O 3  1O 4  1O 2 SNP20
 1O-1 39  6I 1  4I 2  2O 9  2O 6  2O 2  3O 2  1O 1  2O10  6--1 49  4-13O 5 SNP21
 1O 1  2O 1  7O 2 10--1 37  9-17  8O 1  2O-1 -1 49  1L 1O 1V 1E 1  1I 1SSNP22
 1  1A 1  1W 1E 1T 1  1P 1U 2P 1Y 1  1D 1O 1G-1                           SNP23
-2                                                                        *EOP*
 1151 10X-1 51 14X-1 49  4X 4  3X 3  3X-1 48  4X 4  1X 2  2X 3  3X-1 47  3XDOC 1
15  3X-1 46  2X19  2X-1 45  3X20  2X-1 44  3X22  2X-1 44  3X22  3X-1 44  3XDOC 2
14  2X 1  3X 3 11X-1 44  2X12  5X 3  4X 2  5X 3  4X-1 44  2X11  2X24  3X-1 DOC 3
44  2X39  2X-1 44  1X41  2X-1 44  1X42  2X-1 43  2X43  2X-1 43  2X 1  1X 2 DOC 4
 2X38  2X-1 43  1X 1  2X 1  4X38  2X-1 43  1X 2  2X 1  4X23  5X 9  2X-1 43 DOC 5
 1X 2  7X21  8X 8  2X-1 43  1X 2  8X20  8X 8  2X-1 43  1X 2  9X20  6X 9  2XDOC 6
-1 43  1X 2 10X33  2X-1 43  1X 2 10X33  2X-1 43  1X 2 10X32  3X-1 43  2X 1 DOC 7
10X31  3X-1 44  1X 1  8X 1  1X31  2X-1 44  1X 1  8X 1  2X28  3X-1 44  1X 2 DOC 8
 7X 1  3X10 19X-1 45  1X 1  7X 1  4X 8  6X-1 45  1X 2  6X 1  5X 6  2X 2  3XDOC 9
-1 46  1X 1  6X 1  2X 2  2X 5  2X 2  3X-1 46  2X 1  5X 1  2X 3  2X 4  2X 2 DOC10
 3X-1 48  1X 2  2X 2  1X 5  8X-1 49  2X 3  2X 4 10X-1 50  5X 6  2X 6  2X-1 DOC11
52  1X 7  2X 8  1X-1 56  4X 4 10X-1 55  3X 3  6X 3  6X-1 54  2X16  3X 1  1XDOC12
-1 36 19X18 24X-1 35  1X17  4X 3  8X 4  3X22  1X-1 34  1X19  7X 5  3X 2  3XDOC13
24  1X-1 33  1X34  5X26  1X-1 32  1X67  1X-1 31 71X-1 31  1X69  1X-1 31  1XDOC14
52 14X 3  1X-1 31  1X52  1X12  1X 3  1X-1 31  1X 2  4X 2  5X 1  5X 1  5X 1 DOC15
 5X 1  4X 4  3X 2  5X 2  1X 1  3X 2  1X 3  1X 1  1X 3  1X-1 31  1X 2  1X 3 DOC16
 1X 1  1X 3  1X 1  1X 7  1X 3  1X 3  1X 1  1X 3  1X 4  1X 3  1X 6  1X 2  1XDOC17
 3  2X 2  1X 1  1X 3  1X-1 31  1X 2  1X 3  1X 1  1X 3  1X 1  1X 7  1X 3  1XDOC18
 3  1X 1  4X 5  1X 3  5X 2  1X 2  1X 3  1X 1  1X 1  1X 1  1X 3  1X-1 31  1XDOC19
 2  1X 3  1X 1  1X 3  1X 1  1X 7  1X 3  1X 3  1X 1  1X 1  1X 6  1X 7  1X 2 DOC20
 1X 2  1X 3  1X 2  2X 1  1X 3  1X-1 31  1X 2  4X 2  5X 1  5X 3  1X 3  5X 1 DOC21
 1X 2  1X 4  3X 2  5X 2  1X 1  3X 2  1X 3  1X 1  1X 3  1X-1 31  1X52  1X12 DOC22
 1X 3  1X-1 31  1X52 14X 3  1X-1 31  1X69  1X-1 31  1X69  1X-1 31  1X69  1XDOC23
-1                                                                        DOC24
-2                                                                        *EOP*
 11-1 35  6$ 9  5$-1 33 10$ 5 10$-1 32 12$ 3 13$-1 31 14$ 1 15$-1 30 15$ 1 MAY 1
16$-1 30 15$ 1 16$-1 30 15$ 1 16$-1 31 10$ 8 12$-1 32  7$12  8$-1 33  5$14 MAY 2
 5$-1 56  8$-1 29  7$18 11$-1 27  9$17 12$-1 26 11$15 13$-1 26 12$13 14$-1 MAY 3
26 14$ 1  2$ 5  4$ 1 12$-1 27 12$ 1 13$ 1 10$-1 28  9$ 3 13$ 3  6$-1 30  6$MAY 4
 3 14$-1 39 13$ 1  2$-1 40 11$ 2  2$-1 42  9$ 3  2$-1 54  2$-1 55  2$-1 55 MAY 5
 2$-1 55  2$-1 55  2$47  1, 3  1.-1 55  2$47  1, 2  1/ 3  1.-1 55  2$ 7  8$MAY 6
32  1, 1  1/ 2  1. 1- 2  1.-1 54  2$ 4 14$30  1. 2  1. 1- 2  1. 1--1 53  2$MAY 7
 3 16$28  1. 1/ 4  1. 2- 1  3.-1 28  6$18  2$ 2 19$18  1. 8- 1  1* 5  1. 2-MAY 8
```

```
-1 24 14$13  2$ 1 11$27  1/16  7.-1 23 16$11 12$30  1,17  2.-1 22  8# 1  9$MAY 9
 9 10$34  8.11  4--1 21  6$ 7  7$ 6  5$49  1. 8  2- 2.-1 21  2$12  7$ 4  3$MAY10
53  1. 2  1. 3- 2. 3  1--1 36  6$ 3  2$55  1. 1  1, 6  1- 1.-1 37  5$ 2  2$MAY11
55  1/ 3  1,-1 39  4$ 1  2$54  1/ 1  1, 1  1, 1  1,-1 39  4$ 1  2$53  1, 2 MAY12
 1, 1  1, 2  1,-1 40  5$54  1, 2  1, 2  1, 1  1,-1 41  4$54  1, 2  1, 2  1,MAY13
 1  1,-1 42  3$54  1, 3  1, 1  1, 1  1,-1 43  2$55  1, 3  1, 2  1,-1 44  1$MAY14
56  1- 2. 1, 1.10- 1.-1 44  1$59  1I 3 10--1 44  1$53  9/ 1.-1 -1 -1 -1 -1 MAY15
-1 -1 -1 65  1S 1  1I 1  1G 1  1H 2  1. 1  1. 1  1.-1                      MAY16
-2                                                                         *EOP*
 11-1O 1+-1  1 -1O 1+-1  1 -1O 1+63  7$-1  1 -1O 1+59 15$-1  1 -1O 1+57 18$SKP 1
-1  1 -1O 1+57 18$-1  1 -1O 1+35  4$20 15$-1  1 37O 1 -1O 1+33  4$ 1  2$28 SKP 2
 5$-1  1 36O 1 -1O 1+30  6$ 1  3$26  6$-1  1 35O 1 -1O 1+29  6$ 1  3$24  8$SKP 3
-1  1 34O 1 -1O 1+27  7$ 1  3$14  6$ 3  6$ 7 13$-1  1 74O12 -1O 1+26 11$11 SKP 4
16$ 8  2$12  4$-1  1 5OO12 1OO17 -1O 1+25 11$11  2$13 1O$17  2$-1  1 26O 1 SKP 5
2OO 8  2O33 -1O 1+23  3$ 1  8$1O  2$ 8  2$33  2$-1  1 25O 1 2OO 8  2O35 -1OSKP 6
 1+22  3$ 1  8$1O  2$ 8  2$35  2$-1  1 25O 1 19O 8  2O21  9O 7 -1O 1+21  4$SKP 7
 1  6$11  2$ 8  2$21  9$ 7  2$-1  1 24O 1 19O31 12O 6 -1O 1+2O  4$ 1  6$1O SKP 8
 3$31 12$ 6  2$-1  1 45O16  2O12 12O 6 -1O 1+2O 25$16  2$12 12$ 6  2$-1  1 SKP 9
45O16  2O3O -1O 1+2O 25$16  2$3O  2$-1  1 45O17  2O21  2O 5 -1O 1+21 24$17 SKP1O
 2$21  2$ 5  2$-1  1 45O18  2O19  2O 5 -1O 1+23 15$ 4  3$18  2$19  2$ 5  2$SKP11
-1  1 45O2O  2O14  3O 4 -1O 1+43  2$2O  2$14  3$ 4  3$-1  1 46O2O  2O 4 -1OSKP12
 1+44  2$2O  2$ 4 16$-1  1 5OO17 -1O 1+46  4$17  7$-1  1 54O14 -1O 1+49  5$SKP13
14  3$14  4$-1  1 58O1O 18O 2 -1O 1+53  5$1O  2$14  2$ 2  1$ 3  3$-1  1 6OOSKP14
 8 18O 2 -1O 1+58  2$ 8  2$14  2$ 2  9$-1  1 61O 7 18O 2  1O 7 -1O 1+59  2$SKP15
 7  2$14  2$ 2  1$ 7  3$-1  1 84O11 -1O 1+61 1O$11  2$11  4$-1  1 82O14 -1OSKP16
 1+48  4$ 9 12$ 6  3$14  3$-1  1 48O 4 28O 7 -1O 1+46  2$ 4  1$ 7 13$ 4  3$SKP17
 7 12$-1  1 47O 5 14O 6  6O 6 -1O 1+45  2$ 5  1$ 8  5$ 6  6$ 6  4$ 4  5$-1 SKP18
 1 48O 4 14O 8  1O 7 -1O 1+46  2$ 4  2$11  1$ 8  1$ 7  3$-1  1 5OO 2 14O12 SKP19
-1O 1+48  2$ 2 14$12  4$-1  1 49O3O -1O 1+45  4$3O  2$-1  1 45O35 -1O 1+43 SKP2O
 2$35  2$-1  1 45O 8 14O14 -1O 1+42  3$ 8 14$14  2$-1  1 47O 6 14O15 -1O 1+SKP21
42  5$ 6  2$1O  2$15  2$-1  1 46O 3  1O 3 14O16 -1O 1+43  3$ 3  1$ 3  1$11 SKP22
 2$16  2$-1  1 47O 2 18O17 -1O 1+45  2$ 2  5$11  2$17  2$-1  1 66O19 -1O 1+SKP23
47  5$12  2$19  2$-1  1 66O 7  7O 5 -1O 1+64  2$ 7  7$ 5  3$-1  1 66O 5  4OSKP24
 4  2O 5 -1O 1+57  3$ 4  2$ 5  4$ 4  2$ 5  2$-1  1 66O 3  2O 8  2O 5 -1O 1+SKP25
55  4$ 4  3$ 3  2$ 8  2$ 5  2$-1  1 57O 5O 4O 1  3O 9  2O 5 -1O 1+54  3$ 5 SKP26
 4$ 1  3$ 9  2$ 5  2$-1  1 54O 1 13O11  1O 6 -1O 1+53  1$ 1  2$ 5  6$11  1$SKP27
 6  7$-1  1 53O 2 11O12  2O 4  3O 4 -1O 1+52  1$ 2  1$ 5  5$12  2$ 4  3$ 4 SKP28
 3$-1  1 52O 2 11O12  2O 2  3O 6 -1O 1+51  1$ 2  2$ 5  4$12  2$ 2  3$ 6  3$SKP29
-1  1 52O 2 11O11  4O 6 -1O 1+49  3$ 2  2$ 5  4$11  4$ 6  4$-1  1 51O 3 1OOSKP3O
1O  5O 3  3O 1 -1O 1+48  3$ 3  2$ 5  3$1O  5$ 3  3$ 1  3$-1  1 48O 1  2O 3 SKP31
1OO 9  3O 7 -1O 1+47  1$ 1  2$ 3  2$ 5  3$ 9  3$ 7  5$-1  1 48O 2  1O 3 1OOSKP32
 7  3O 6 -1O 1+47  1$ 2  1$ 3  2$ 5  3$ 7  3$ 6 12$-1  1 48O 2  1O 4 1OO 3 SKP33
 3O 6  5O 6 -1O 1+46  2$ 2  1$ 4 1O$ 3  3$ 6  5$ 6  3$-1  1 48O 2  1O 6  8OSKP34
 2 13O 4 -1O 1+46  2$ 2  1$ 6  8$ 2 13$ 4  5$-1  1 48O 2 13O 5  6O 6 -1O 1+SKP35
46  2$ 2 13$ 5  6$ 6  9$-1  1 49O 7 14O 5  7O 7 -1O 1+47  2$ 7 14$ 5  7$ 7 SKP36
 5$-1  1 76O16 -1O 1+48 28$16  3$-1  1 65O1O -1O 1+54  4$ 4  3$1O 21$-1  1 SKP37
7OO 7  1O 7 -1O 1+47 1O$ 7  6$ 7  1$ 7  5$-1  1 81O 4 -1O 1+48  8$13 12$ 4 SKP38
 6$-1  1 -1O 1+79 11$-1  1 -1O                                             SKP39
-2                                                                         *EOP*
 1156 2O*-1  1+56 2OQ-1 54  3*2O  3*-1  1+53  3Q2O  3Q-1 52  2*26  2*-1  1+YOY 1
51  2Q26  2Q-1 5O  2*1O 1O* 9  1*-1  1+49  2Q1O 1OQ 9  1Q-1 49  1*22  1* 9 YOY 2
```

```
 2*-1  1+48  1Q22  1Q 9  2Q-1 48  1*23  1*11  1*-1  1+47  1Q23  1Q11  1Q-1 YOY 3
46  2* 8  1* 3  2* 8  2*13  1*-1  1+45  2Q 8  1Q 3  2Q 8  2Q13  1Q-1 46  1*YOY 4
10 13*16  2*-1  1+45  1Q10 13Q16  2Q-1 45  1*41  1*-1  1+44  1Q41  1Q-1 45 YOY 5
 1*42  1*-1  1+44  1Q42  1Q-1 45  1*12  3* 3  4* 2  3*15  1*-1  1+44  1Q12 YOY 6
 3Q 3  4Q 2  3Q15  1Q-1 45  1*12  3* 7  1* 1  3*15  1*-1  1+44  1Q12  3Q 7 YOY 7
 1Q 1  3Q15  1Q-1 45  1*22  1*19  1*-1  1+44  1Q22  1Q19  1Q-1 45  1*16  6*YOY 8
20  2*-1  1+44  1Q16  6Q20  2Q-1 43  3*42  1* 1  2*-1  1+42  3Q42  1Q 1  2QYOY 9
-1 43  1*33  1*10  1* 1  2*-1  1+42  1Q33  1Q10  1Q 1  2Q-1 43  1*33  3* 8 YOY10
 1* 1  2*-1  1+42  1Q33  3Q 8  1Q 1  2Q-1 43  1*10  3*19  1*10  3*-1  1+42 YOY11
 1Q10  3Q19  1Q10  3Q-1 44  4* 9  1*16  2*11  1*-1  1+43  4Q 9  1Q16  2Q11 YOY12
 1Q-1 48  1* 9 16*12  1*-1  1+47  1Q 9 16Q12  1Q-1 49  1*35  1*-1  1+48  1QYOY13
35  1Q-1 50  2*32  1*-1  1+49  2Q32  1Q-1 52  1*29  2*-1  1+51  1Q29  2Q-1 YOY14
53  3*24  2*-1  1+52  3Q24  2Q-1 56  2*18  4*-1  1+55  2Q18  4Q-1 58 18*-1 YOY15
 1+57 18Q-1 58 16*-1  1+57 16Q-1 44  4*10  3*12  3*14  4* 2  2*-1  1+43  4QYOY16
10  3Q12  3Q14  4Q 2  2Q-1 39  5* 2  2* 9  1* 3  1*10  1* 3  1*12  1* 4  2*YOY17
 1  3*-1  1+38  5Q 2  2Q 9  1Q 3  1Q10  1Q 3  1Q12  1Q 4  2Q 1  3Q-1 39  2*YOY18
 2  1* 2  4* 6  1* 5 10* 5  1*11  1* 4  2* 1  3*-1  1+38  2Q 2  1Q 2  4Q 6 YOY19
 1Q 5 10Q 5  1Q11  1Q 4  2Q 1  3Q-1 41  3* 2  2* 2  6*22 12* 3  1* 3  4*-1 YOY20
 1+40  3Q 2  2Q 2  6Q22 12Q 3  1Q 3  4Q-1 39  5* 4  1* 3  1*33  2* 4  1* 7 YOY21
 1*-1  1+38  5Q 4  1Q 3  1Q33  2Q 4  1Q 7  1Q-1 37  2* 2  2* 5 42* 7  1*-1 YOY22
 1+36  2Q 2  2Q 5 42Q 7  1Q-1 37  7* 8  1*27  1* 5  2* 9  3*-1  1+36  7Q 8 YOY23
 1Q27  1Q 5  2Q 9  3Q-1 40  4* 8  1*28  1* 4  2* 5  3* 1  1*-1  1+39  4Q 8 YOY24
 1Q28  1Q 4  2Q 5  3Q 1  1Q-1 43  3* 6  1*29  2* 4  1* 4  1* 2  2*-1  1+42 YOY25
 3Q 6  1Q29  2Q 4  1Q 4  1Q 2  2Q-1 43  1* 1  1* 4  2* 2  2*26  2* 5  1* 2 YOY26
 2*-1  1+42  1Q 1  1Q 4  2Q 2  2Q26  2Q 5  1Q 2  2Q-1 43  1* 2  8*28  2* 5 YOY27
 3*-1  1+42  1Q 2  8Q28  2Q 5  3Q-1 43  1* 8  1* 7  1* 7  1* 9  2* 2  7*-1 YOY28
 1+42  1Q 8  1Q 7  1Q 7  1Q 9  2Q 2  7Q-1 43  1* 8  2* 5  4* 4  4* 6  4* 1 YOY29
 4*-1  1+42  1Q 8  2Q 5  4Q 4  4Q 6  4Q 1  4Q-1 43  1* 8  4* 2  6* 2  6* 4 YOY30
10*-1  1+42  1Q 8  4Q 2  6Q 2  6Q 4 10Q-1 43  1* 8  5* 1  7* 1  7* 1 13*-1 YOY31
 1+42  1Q 8  5Q 1  7Q 1  7Q 1 13Q-1 43  1* 8 35*-1  1+42  1Q 8 35Q-1 43  1*YOY32
 8  1* 1  7* 1  7* 1  7* 3  7*-1  1+42  1Q 8  1Q 1  7Q 1  7Q 1  7Q 3  7Q-1 YOY33
40  6* 6  1* 3  4* 4  4* 4  4* 5  4* 1  1*-1  1+39  6Q 6  1Q 3  4Q 4  4Q 4 YOY34
 4Q 5  4Q 1  1Q-1 38  2* 6  2* 4  1* 4  1* 7  1* 7  1* 8  2* 2  1*-1  1+37 YOY35
 2Q 6  2Q 4  1Q 4  1Q 7  1Q 7  1Q 8  2Q 2  1Q-1 37  1*10  1* 3  1*33  1*-1 YOY36
 1+36  1Q10  1Q 3  1Q33  1Q-1 36 14* 2  1*33  1*-1  1+35 14Q 2  1Q33  1Q-1 YOY37
36  1*12  1* 2 35*-1  1+35  1Q12  1Q 2 35Q-1 36 14* 2 35*-1  1+35 14Q 2 35QYOY38
-1 36 14* 2 35*-1  1+35 14Q 2 35Q-1 37  1*10  1* 3 35*-1  1+36  1Q10  1Q 3 YOY39
35Q-1 38  2* 6  2* 9  1* 3 17* 3  1*-1  1+37  2Q 6  2Q 9  1Q 3 17Q 3  1Q-1 YOY40
40  6*11  1* 8  2* 4  1* 8  1*-1  1+39  6Q11  1Q 8  2Q 4  1Q 8  1Q-1 57  1*YOY41
 3  7* 2 12*-1  1+56  1Q 3  7Q 2 12Q-1 57 11* 1 21*-1  1+56 11Q 1 21Q-1 54 YOY42
11* 3 17* 5  3*-1  1+53 11Q 3 17Q 5  3Q-1 53  1*23  1*14  1*-1  1+52  1Q23 YOY43
 1Q14  1Q-1 52  1*23 17*-1  1+51  1Q23 17Q-1 53 23*-1  1+52 23Q-1  0  0  0 YOY44
-2                                                                        *EOP
 1152  3*-1 51  6*-1 49  1* 8  1*-1 48  1*10  1*-1 48  1*10  1* 7  5*-1 47 DGH 1
 1*11  1* 4  1* 8  1*-1 47  1*10  1* 4  1*10  1* 6  4*-1 47  1*10  2* 2  1*DGH 2
12  1* 3  2* 2  1* 1  1*-1 46  1* 1  2O 6  1* 2  2& 9  1* 3  2* 8  1*-1 45 DGH 3
 1* 8  1* 5  2&14  1* 3  2* 3  1*-1 44  1*15  2&15  3* 1  5*-1 44  1*15  2&DGH 4
 2  9*12  1*-1 44  1*15  2&11  1*11  1*-1 46  2* 2  3* 7  2& 7* 5  6* 5  1*DGH 5
-1 36 12X 2* 3& 7* 2& 7X 1* 2  2* 6X 5*16X-1 36  1X12  5&16  2*28  1X-1 36 DGH 6
 1X12  5&46  1X-1 36  1X12  5&46  1X-1 36  1X11  6&46  1X-1 36  1X11  6&46 DGH 7
 1X-1 35 13X 6&48X-1 35  1X11  7&47  1X-1 35  1X11  7&47  1X-1 35  1X12  6&DGH 8
```

```
47  1X-1 35  1X13  4&48  1X-1 35  1X14  2&49  1X-1 35  1X65  1X-1 34 69X-1 DGH 9
34  1X67  1X-1 34  1X67  1X-1 34  1X67  1X-1 34  1X67  1X-1 34  1X67  1X-1 DGH10
34  1X67  1X-1 34 69X-1 35  1X65  1X-1 35 67X-1 43  1X47  1X-1 43  1X47  1XDGH11
-1 43 49X-1 43  1X47  1X-1 43  1X47  1X-1 43  1X47  1X-1 43  1X47  1X-1 43 DGH12
 1X47  1X-1 43 49X-1 43  1X47  1X-1 43  1X47  1X-1 43  1X47  1X-1 43  1X47 DGH13
 1X-1 43  1X47  1X-1 43 49X-1 43  1X47  1X-1 43 49X-1                      DGH14
-2                                                                         *EOP*
 11   89 11*-1 87 19*-1 84  4* 4  2* 3  2* 2  2* 2  4*-1 81  3* 9 14* 1  3*PNT 1
-1 28  4*47  3*23  2* 3  3*-1 27  2*47  3*28  4* 1  3*-1 26  2*48  3*30  2*PNT 2
 3  2*-1 19  4* 3  1*17 12*20  2*33  3* 1  2*-1 16 12*14  4* 7  6*17  2* 3 PNT 3
 3*32  2*-1 14  5* 8  4* 1  2* 5  4*15  3*15  3* 3  5*30  2*-1 12  3*14 12*PNT 4
20  3* 1  3* 9  3* 6  4*27  3*-1 11  3*18  2*28  7* 9  2* 9  4*25  2*-1 10 PNT 5
 2*20  2*28  3* 2  3* 9  2*11  5*21  3*-1  9  2*21  2*29  7* 9  3*14  5*18 PNT 6
 2*-1  9  1*23  2*28  7*10  3*17  5*14  2*-1  8  1*53  6*13  3*19  6*10  2*PNT 7
-1  8  1*53  2*19  3*22  7* 4  2*-1  7  3*20  6*24  3*21  4*26  3* 2  2*-1 PNT 8
 7  3*18 13*16  5*24  4*28  2*-1  7  5*16  2* 8 21*29  4*24  3*-1  7  7*14 PNT 9
 2*16  9*36  6*16  4*-1  7  8*13  2*64  7* 7  6*-1  6 11*11  3*69 10*-1  6 PNT10
12*11  2*-1  6 14* 9  3*-1  6 15*10  3*-1  6 15*11  5* 8  6*-1  7 15*13 12*PNT11
 2  2*-1  7 16*24  3*-1  8 16*16  9*-1  9 16*15  2* 2  2*-1 10 21*10  6* 4 PNT12
 9*18  6*-1 11 24* 5 24*13  2* 4  3*-1 12 15* 6 10*19  5*10  2* 5  3*-1 13 PNT13
15* 7  5* 8  7*11  3* 8  2* 3  2* 2  2*-1 15 13* 9  1* 5 16* 9  3* 6  2* 8 PNT14
 2*-1 17 11* 7  3* 1  5*14  3* 9  2* 6  2* 5  5*-1 20  8* 6  3*23  2* 9  2*PNT15
 6  2* 8  2*-1 22  6* 5  2*25  2* 9  3* 7  2* 6  2*-1 24  3* 6  2* 8  8* 3 PNT16
 1* 2  1* 3  2* 9  2* 8  2* 5  3*-1 33 10* 1  2* 4  2* 2  1* 2  1* 2  3* 9 PNT17
 2* 8  2* 6  2*-1 45  1* 5 11*10 12* 6  2*-1 45  2*25  5* 4  3* 6  2*-1 45 PNT18
 2*43  2*-1 46  1*43  2*-1 46  2*42  2*-1 47  2*26  8* 7  2*-1 48  2*24 13*PNT19
 2  3*-1 49  2*21  3* 9  6*-1 50  2*18  3*-1 51  4* 6  2* 7  2*-1 53 10* 6 PNT20
 2*-1 59  3* 6  2*-1 49 11* 4  9*-1 47  3*22  2*-1 47  2*24  2*-1 48  2*15 PNT21
 3* 6  1*-1 49  3* 7  2* 6  3* 2  2*-1 51  4* 5  2* 7  3*-1 54 16*-1 -1 -1 PNT22
-2                                                                         *EOP*
 1162 10*-1 59  4*10  4*-1 56  3*18  2*-1 54  2* 6  6*11  2*-1 53  1* 6  2*CHB 1
19  1*-1 52  1* 6  1*22  1*-1 51  1* 7  1* 6  1* 6  1* 9  1*-1 50  1* 9  2*CHB 2
 2  2* 1  2* 2  2*10  1*-1 50  1*11  2* 5  2*12  1*-1 49  1* 6  1*16  2* 9 CHB 3
 1*-1 49  1* 5  1* 7  4* 8  1* 8  1*-1 48  1* 5  1* 4  2O 1  1* 5  2O 5  1*CHB 4
 8  2*-1 48  1* 5  1* 7  1*11  1* 8  1* 2  1*-1 48  1* 6  1* 7  3*21  1*-1 CHB 5
48  1*37  1*-1 49  1*33  3*-1 50  1*32  1*-1 50  1* 7  1I14- 1I 8  1*-1 51 CHB 6
 1* 1  1* 2  2*23  1*-1 52  1* 1  2* 2  3*18  2*-1 52  1* 1  1* 2  2* 2  1*CHB 7
14  3*-1 52  1* 5  4*10  4*-1 53  1* 5 13* 3  1*-1 54  1* 5  2*12  2*-1 53 CHB 8
 2* 5 13* 3&-1 50  2* 8  3* 4& 4* 7&-1 52  2* 6  4* 2& 6* 5&-1 49  1*10  1*CHB 9
17&-1 48  1*11  1*18&-1 48  1*10  1*20&-1 47  1& 1* 9  1*21&-1 47  2& 1* 7 CHB10
 1*13& 1* 8&-1 47  3& 2* 2  3*14& 2* 8&-1 48  2& 1O 1& 2* 3& 1O 6& 1O 6& 1OCHB11
 1* 8&-1 49  3O 4& 3O 4& 3O 4& 2O 1* 8&-1 49  4O 2& 5O 2& 5O 2& 3O 1* 6  1*CHB12
 1&-1 49 23O 1* 6  1* 2&-1 49  2& 6O 1& 6O 1& 6O 1& 1* 6  1* 2&-1 48  4& 4OCHB13
 3& 4O 3& 4O 2& 1* 6  2* 2&-1 48  5& 2O 5& 2O 5& 2O 3& 1* 1  1* 1  1* 1  4*CHB14
 1&-1 48 24& 1* 1  1* 1  2* 1  1* 3&-1 48 24& 5* 1& 1* 4&-1 49 33X-1 49 33XCHB15
-1 49 33X-1 57  1* 7  1* 2  1* 8  1*-1 57  9* 2  8*-1 57  1* 7  1* 2  1* 8 CHB16
 1*-1 52 28*-1 48  4* 9  4*14  2*-1 46  2*12  2*18  2*-1 45 37*-1 79 -1 54 CHB17
 1O 1H 1, 1  1G 2O 1D 1  1G 1R 1I 1E 1F-1                                  CHB18
-2                                                                         *EOP*
 11-1  1 -1  1 -1 57 13X-1 54  4X11  3X-1 54  1X20  2X-1 53  2X22  2X-1 52 SHR 1
 2X26  2X-1 51  2X29  2X-1 50  2X32  2X-1 49  2X34  2X-1 49  2X35  2X-1 49 SHR 2
```

```
 2X36  2X-1 49  2X16  5X16  2X-1 37 11X 1  2X 9  1X 4  1X 2  3X 2  1X15  2XSHR 3
-1 33  4R 8  3X 2  1X 7  2X 4  1X 1  1X 5  1X 1  1X14  2X-1 29  9R 7  5X 1 SHR 4
 6X 1  2X 1  2X 1  1X 6  1X 2  1X13  2X-1 26 12R 7  3X17  1X 6  1X 1  1X13 SHR 5
 2X-1 24 15R 6  3X23  1X 1  1X12  2X-1 21  1X 4 14R 5  3X21  1X 1  2X11  2XSHR 6
-1 19  1X 7 14R 4  7X19  2X 9  2X-1 17  1X10 13R 2  5X 3  1X22  1X 5  2X-1 SHR 7
15  2R12  7R 1X 8  3X 4  1X23  1X 4  2X-1 14  4R12  5R 1X 9  2X 5  1X14  2XSHR 8
 7  1X 5  2X-1 13  6R12  3R 1X10  1R 1X 6  1X14  2X 6  1X 6  2X-1 12  8R12 SHR 9
 2R 1X 7  4R 1X 7  1X15  1X 1  4X 8  2X-1 11 10R12  1R 1X 4  7R 1X 8  1X14 SHR10
 2X 3  1X 8  2X-1 10 12R 9  4V 1X10R 1X 9  1X13  1X 4  1X 8  2X-1 10 13R 5 SHR11
 9V 8R 2X11  1X 5  2X 4  1X 5  1X 6  2X-1  9 15R 2 13V 4R 3X 1  1X12  1X 1 SHR12
 1X 2  1X 4  1X 6  1X 4  2X-1  8  1X 2 13R16V 1R 3X 4  1X17  3X 9  3X-1  8 SHR13
 1X 4 10R17V 2X 7  2X-1  7  1X 7  7R18V11  2X-1  7  1X10  3R19V 1X12  1X 5-SHR14
-1  6  1X12  1R20V 1  2X12  4- 9  1/-1  6  2R12 20V 3  3X 9  4- 8  1/-1  6 SHR15
 4R 9 20V 4  3B 1X 7  4- 8  1/-1  6  6R 6 21V 3  5B 1  4X 2  4- 1 590-1  6 SHR16
 9R 3 20V 3  6B 6  3- 3 59O-1  7 10R20V 3  7B12 59O-1  7 10R20V 1X 1  8B 9 SHR17
 1%61O-1  7  1X 2  7R19V 2  9B 8  1%62O-1  8  1X 7 19V 2  9B72O-1  9  1X 6 SHR18
18V 3  9B72O-1 10  1X 5 17V 1X 2  9B40O 1/24  3O-1 11  1X 4 16V 2  1X 1  9BSHR19
 8  3O54  3O-1 12  1X 3 15V 4  9B 9  3O54  3O-1 13  1X 2 13V 6  8B10  3O54 SHR20
 3O-1 14  2X11V 8  7B11  3O54  3O-1 17  7V11  5B12  3O54  3O-1 36  3B-1    SHR21
-2                                                                        *EOP*
 11-1 54  1/ 4- 1X 6  1/ 4- 1X-1 53  1/ 1  4- 1  1X 4  1/ 1  1/ 2- 1X 1  1XPLT 0
-1 52  1/ 1  1/ 4  1X 1  1X 2  1/ 1  1/ 4  1X 1  1X-1 51  1/ 1  1/ 2  1/ 3-PLT 1
 1X 1  1I 1- 1I 1  1I 1X 3  1I 1  1I-1 51  1I 1  1I 1  1/-1 51  1I 1  1I 1 PLT 2
 1/ 4O 1I 1  1I 1- 1I 1  1I 1O 1X 2  1I 1  1I-1 51  1I 1  1I 1/ 4O 1/ 1  1/PLT 3
 3O 1X 1  1X 1O 1X 1  1/ 1  1/-1 51  1I 1  1X 1/ 3O 1/ 1  1/ 1O 1/ 3- 1X 1 PLT 4
 1X 1- 1X 1  1/-1 52  1X 1  1X 2- 1/ 1  1/ 1O 1/ 5  1X 3- 1/-1 51  1/ 1S 1XPLT 5
 2- 1/ 1- 1X 1O 1/10  1X-1 49  1/ 1X 2S 1/ 5O 1/12  1X-1 48  1/ 2  1X 1/ 5OPLT 6
 1/14  1X-1 47  1/ 4  1X 4O 1I 6  2/ 1X 3  2/ 1X 1  1X-1 47  1I 5  1X 3O 1IPLT 7
 5  2/ 2  1X 1  2/ 2  1X 1  1X 6- 4& 3- 1X-1 47  1I 6  1I 2O 1I 6  1% 18 1<PLT 8
 3  1% 18 1<17  1X-1 47  1I 2  1& 3  1I 1O 1% 1/ 1<32  1X-1 47  1I 6  1I 2OPLT 9
 1I34  1X-1 47  1X 5  1/ 3O 1I35  1X-1 48  1X 3  1/ 4O 1I36  1X-1 49  1X 1-PLT10
 1/ 5O 1X28  1X 1- 1X 5  1<-1 50  1X 7O 1X26  1% 3& 1< 4  1<-1 51  1X 7O 1XPLT11
25  1% 3& 1< 4  1<-1 52  1X 7O 1X 2- 1X23  1X 1- 1X 5  1<-1 53  1X10O 1X 7 PLT12
 3*19  1<-1 54  1X 9- 1X10  1*18  1/-1 63  2X11  1*16  1/-1 65  1X11  1* 2-PLT13
 1X11  1/-1 66  1I 7  1/ 3* 3  1X 9- 1/-1 40  1* 1# 1* 1# 1* 1# 1* 7- 1*11 PLT14
 1I 6  1/-1 41  1* 1I11  1I 1* 5- 1* 4  1I 6  1I10  2--1 40  1* 1I11  1I 1*PLT15
 5  1I 1* 1# 1* 1< 1* 1# 1* 1# 1* 1# 1* 1< 9  1< 2O 1<-1 41  1* 1I11  1I 1*PLT16
 3  1* 1# 1* 3  1% 7O 1< 6  1- 2  1% 2O 1<-1 41  1* 1I11  1I 5- 2* 1# 1* 1%PLT17
 1* 1# 1* 1# 1* 1# 1* 1< 5& 1/ 1  2X 1, 5&-1 38  1* 1# 1* 1# 1*11- 1* 3- 1*PLT18
 6  1I 7  1I 3& 2- 1< 6  1% 3  1<-1 64  1/22  1< 1  1%-1 63  1/22  1< 1  1%PLT19
-1 62  1/18  1- 4  1< 1  1%-1 61  1/ 1* 9  1/ 3& 1I 4& 1/ 1  1/ 2& 1/ 2& 1/PLT20
-1 60  1/ 2&13  1I 8  2X-1 59  1/ 3&13  1I 1  1/ 3& 1< 2  2X-1 58  1/ 3&13 PLT21
 1/ 3& 1< 2  1I 2  2X-1 43  1-13  1/ 4&12  1/ 4& 1I 2  1/ 2  2X-1 45  1-10 PLT22
 1/ 4*13  1I 4  1/ 2  1X 2  2X-1 44  1- 1  1- 7  1/ 3*15  1X 4  1X 3  1< 1 PLT23

 2X-1 45  1- 3  1- 1  1- 1  1  1%11  1/ 5& 1X 1  1X 4  1< 2  1I 1  2X-1 47 PLT24
 1- 1  4* 1  1%10  1& 7  1X 5  1I 2  1I 1  2X-1 50  1- 1  1- 1  1- 1X23  1IPLT25
 2  1I 1  2X-1 56  1X22  1I 2  1I 1  2X-1 57  1X21  1I 2  1I 1  2X-1 58  1XPLT26
 6  1/ 7& 1X 5  1I 2& 1< 1/ 2X-1 59  1X 4& 1/ 9  1X 4& 1< 2  1/ 2X 1/-1 45 PLT27
45X-1 45 45X-1                                                            PLT28
-2                                                                        *EOP*
```

```
 11-1                                                                   LHP00
82  1- 6X 1--1 78  4X 8  2X-1 56  5X16  1X15  2X-1 54  2X 5  2X13  1X18  2XLHP 1
-1 52  2X 9  3X 9  1X21  2X-1 50  2X10  1* 1- 2  3X 5  1X15  4* 5  2X-1 49 LHP 2
 2X 9  1/ 3* 5  5X15  6* 6  1X-1 48  3X 8  1- 2*27  7* 6  1X-1 47  3X40  5*LHP 3
 7  1X-1 47  2X18  1/ 1<33  1X-1 46  3X18  1/34  1X-1 46  3X17  1% 1/ 1  1XLHP 4
32  1X-1 44  3X 1  1X20  1- 2X29  1X-1 43  4X 1  1X22  9X20  1X-1 42  4X 2 LHP 5
 1X28  8X13  2X-1 41  1X 1  3X 1  2X26  2X 8 13X-1 41  4X 1  1X26  2X-1 39 LHP 6
 5X 2  2X23  2X-1 39  5X 1  1X 1  2X21  2X-1 38  6X 1  1X 3  2X18  3X-1 37 LHP 7
 4X 1  3X 7  1X14  6X-1 37  4X 1  2X10  1X11  1X 2  5X-1 37  4X 1  2X11  1XLHP 8
 9  1X 3  5X-1 36  3X 1  4X12  1X 7  1X 5  6X-1 36  3X 1  4X13  1X 6  1X 6 LHP 9
 6X-1 36  7X14  1X 6  1X 7  5X-1 36  8X13  1X 6  1X 7  5X-1 37  8X12  1I 6/LHP10
 1I 8  5X-1 38  7X12  1I 6/ 1I 8  5X-1 38  8X11  1I 6/ 1I 8  5X-1 39  7X11 LHP11
 1X 6  1X 8  5X-1 41  5X11  1X 6  1X 8  5X-1 57  1X 6  1X 8  4X-1 57  1X 6 LHP12
 1X 8  3X-1 57  1X 6  1X-1 56  1X 7  1X-1 55  1X 9  1X-1 54  2X 9  1X-1 54 LHP13
 1X10  1X-1 53  1X11  1X-1 53  1X11  1X-1 46  6X 1  1X12  1X 2  6X-1 44  3XLHP14
 4  2X13  1X 1  2X 5  2X-1 43  2X 7  2X13  1X 8  2X-1 43  1X10  1X11  2X 9 LHP15
 1X-1 42  1X11  1X11  1X11  1X-1 42  1X11  1X11  1X11  1X-1 43  1X10  1X11 LHP16
 1X11  1X-1 43  1X10  1X11  1X11  1X-1 44  1X 9  2X10  1X10  1X-1 44  2X 8 LHP17
 1X11  1X10  1X-1 34  8X 4  1X 7  2X 3  2X 5  1X 9  1X-1 33  2X 7  4X 1  1XLHP18
 3  4X 4  2X 5  1X11  7X-1 31  2X 4  2X 7  6X 7  2X 6  2X 6  1X 8  3X-1 31 LHP19
 1X 4  2X21  2X 7  1X 6  2X 5  1X 4  1X-1 31  1X 4  1X 5  1X 6  1X 9  2X 8 LHP20
 2X 6  1X 4  1X 4  1X-1 31  1X 3  1X 6  1X 5  1X10  2X10  1X 6  2X 3  1X 3 LHP21
 1X-1 31 57X-1                                                          LHP22
-2                                                                      *EOP*
 11-10 1 -10 1 -1075  9*-1  1+-1071  4* 7  3*-1  1+740 7 -1069  2*14  1*-1 BGL 1
 1+70014 -1066  2* 3  3*12  1*-1  1+670 3  3012 -1050 15* 3  1* 5  1*12  1*BGL 2
-1  1+640 3  10 5  1012 -1047  3*18  1* 5  1*13  1*-1  1+49018  10 5  1013 BGL 3
-1045  2*20  2* 4  2*13  1*-1  1+46020  20 4  2013 -1044  1*22  2* 4  2*14 BGL 4
 1*-1  1+44022  20 4  2014 -1043  1*38  6* 2  1*-1  1+43038  60 2 -1042  1*BGL 5
12  6*19  9* 1  1*-1  1+42012  6019  90 1 -1042  1*11  8*17 10* 2  1*-1  1+BGL 6
42011  8017 100 2 -1042  1*12  5*18 12* 1  1*-1  1+42012  5018 120 1 -1042 BGL 7
 1*34 13* 1  1*-1  1+42034 130 1 -1042  1*28  1* 5 13* 1  1*-1  1+42028  10BGL 8
 5 130 1 -1043  1* 7  1*18  1* 1  1* 3 14* 1  1*-1  1+430 7  1018  10 1  10BGL 9
 3 140 1 -1044  1* 5  1* 1  1*16  1* 2  1* 3 14* 1  1*-1  1+440 5  10 1  10BGL10
16  10 2  10 3 140 1 -1045  2* 3  1* 2  3*11  2* 1  1* 1  1* 2 15* 1  1*-1 BGL11
 1+460 3  10 2  3011  20 1  10 1  10 2 150 1 -1047  4* 1  1* 3 12* 3  2* 2 BGL12
14* 2  1*-1  1+500 1  10 3 120 3  20 2 140 2 -1051  3* 2  1* 4  1* 2  1* 3 BGL13
 3* 4 14* 1  1*-1  1+530 2  10 4  10 2  10 3  30 4 140 1 -1054 15* 4 15* 1 BGL14
 1*-1  1+680 4 150 1 -1062  1* 8  2* 2 12* 2  1*-1  1+620 8 160 2 -1063  1*BGL15
 6  1* 5 10* 1  2*-1  1+630 6 160 1 -1061 10* 6 10*-1  1+-1061 10* 7  6*-1 BGL16
 1+-1059  2* 8  1*-1  1+600 8 -1058  1*11  1*-1  1+58011 -1056  2* 7  1* 5 BGL17
 1*-1  1+570 7  10 5 -1055  2* 8  1* 5  1*-1  1+560 8  10 5 -1054  2* 8  1*BGL18
 7  1*-1  1+550 8  10 7 -1053  2* 9  1* 4  1* 2  1*-1  1+540 9  10 4  10 2 BGL19
-1053  1* 9  1* 5  1* 3  1*-1  1+530 9  10 5  10 3 -1052  1*10  1* 5  1* 3 BGL20
 1*-1  1+52010  10 5  10 3 -1052  1*10  1* 5  1* 3  1*-1  1+52010  10 5  10BGL21
 3 -1052  1*10  1* 4  1* 1  2* 2  1*-1  1+52010  10 4  10 1  20 2 -1051  1*BGL22
10  1* 5  2* 2  1* 1  1*-1  1+51010  10 5  20 2  10 1 -1051  1*10  1* 5  1*BGL23
 3  1* 2  1*-1  1+51010  10 5  10 3  10 2 -1051  1*10  1* 7  3* 2  1*-1  1+BGL24
51010  10 7  30 2 -1051  1*10  1* 9  1* 3  1*-1  1+51010  10 9  10 3 -1051 BGL25
 1*10  1* 6  1* 1  1* 4  1*-1  1+51010  10 6  10 1  10 4 -1052  1*10  1* 5 BGL26
 2* 5  1*-1  1+52010  10 5  20 5 -1052  1*11  5* 7  1* 3  2*-1  1+52011  50BGL27
```

```
 7 -1O53  1*21  1* 3  1* 2  1*-1  1+53O21  5O 2 -1O54  2*18  5* 3  1*-1  1+BGL28
55O18  5O 3 -1O56  1*25  1*-1  1+56O25 -1O47  6* 4  2*22  1*-1  1+58O22 -1OBGL29
46  1* 6  3* 2  1* 3  9* 9  1*-1  1+46O 6  6O 3  9O 9 -1O46  1* 2  1* 6  1*BGL30
 1  1* 3  1* 6  1* 9  1*-1  1+46O 2  1O 6  3O 3  8O 9 -1O47  2* 2  1* 5  1*BGL31
 4  1* 5  1* 5  1* 3  1*-1  1+48O 2  1O 5  1O 4  7O 5  1O 3 -1O48  3*11  2*BGL32
 4  1* 2  1* 2  1* 1  2*-1  1+5OO11  7O 2  1O 2  1O 1 -1O5O  2*12  1* 3  1*BGL33
 2  1* 2  1*-1  1+51O12  5O 2  1O 2 -1O52  3*10  1* 2  1* 2  3*-1  1+54O1O BGL34
 4O 2 -1O55  4* 6  1* 3  2*-1  1+58O 6 -1O59  6*-1  1+-1O             1 -1OBGL35
5O 32--1  1+-1O33 63--1  1+-1O42 67--1  1+-1O59 44--1  1+-1O 1 -1O         BGL36
-2                                                                        *EOP*
 11-1                                                                     LNS00
58  7*-1 53  4* 8  3*-1 51  2*16  1*-1 5O  2* 7  1*1O  1* 9  1*-1 49  1*21 LNS 1
 2* 3  1* 1  1* 2  1*-1 49  1* 7  1*15  1* 1  2* 5  1*-1 48  1*1O  2*13  1*LNS 2
 7  1*-1 49  1*23  1* 8  1*-1 48  2* 4  1* 5  3* 4  1* 4  1* 9  1*-1 47  1*LNS 3
 1  1* 3  1* 2  2O 1  1* 4  2O 2  1* 2  1*1O  1*-1 47  2* 4  1* 5  1* 8  1*LNS 4
 1  1*11  1*-1 47  1* 6  1* 5  2* 5  1* 2  1* 5  2* 4  1*-1 46  1*22  1* 4 LNS 5
 3* 1  1* 3  1*-1 46  1* 9  3* 2  2* 6  1* 2  3* 1  1* 1  1* 3  1*-1 46  1*LNS 6
 8  2* 1  3* 2  1* 4  1* 2  2* 2  1* 2  1* 2  1*-1 47  1* 6  2* 1  2* 1  1*LNS 7
 2  1* 4  1* 2  1* 2  1* 3  1* 2  1*-1 5O  1* 3  3* 3  1* 6  1* 3  1* 3  1*LNS 8
 1  1* 2  1*-1 49  6* 5  1* 6  1* 2  1* 3  1* 2  1* 1  1*-1 48  1*11  8* 1 LNS 9
 1* 5  1* 1  1* 1  1*-1 47  1*11  1* 6X 3* 7  2* 1  1*-1 46  2*11  1* 6X 1*LNS1O
 1  1* 8  2*-1 45  1* 2O 1* 9  1* 7O 1* 1  1* 8  1*-1 45  1* 3O 1* 7  1* 8OLNS11
 1* 1  1* 8  1*-1 45  1* 3O 1X 7* 8X 1* 2  1* 6  1* 1O-1 45  1* 4O15X 1* 1 LNS12
 1* 1  2* 2  2* 1O-1 46  2*17O 1* 1  1* 3  1* 3O 1*-1 48  1*16O 1* 1  1* 3 LNS13
 1* 3O 1*-1 48  1*16O 1* 1  1* 4  1* 1O 1*-1 48  1*16X 1* 1  1* 4  1* 1X 1*LNS14
-1 48  1*15X 1* 2  1* 4  1* 1X 1*-1 47  1*15O 1* 2  1* 4  1* 1O 1*-1 46  1*LNS15
17O 1* 2  1* 4  1* 1O 1* 6  2*-1 45  1*18X 1* 2  1* 4  1* 1X 1* 4  1* 3  1*LNS16
-1 45  1*18X 1* 2  1* 4  1* 2X 1* 2  1* 4  1*-1 45  1*18# 1* 2  1* 4  1* 2#LNS17
 1* 1  1* 4  1*-1 45  1*19# 1* 1  1* 4  1* 1# 1* 1  1* 5  1*-1 45  1*19# 1*LNS18
 2  1* 3  1* 1# 2* 2  1* 3  1*-1 45 21* 3  1* 3  2* 2  2* 3  1* 3  5*-1 52 LNS19
 1* 7  1* 1  1* 3  1* 2  1* 6  2* 3  1* 1  3* 5  1*-1 52  1* 7  1* 1  1* 3 LNS2O
 1* 3  1* 4  1* 1  1* 3  2* 7  1* 3  3*-1 52  9* 1  6* 3  1* 2  1* 2  1*11 LNS21
 5* 2  5*-1 52  1* 7  1* 1  1* 5  1* 2  1* 2  1* 2  1*23  1*-1 45 25* 2  2*LNS22
 2  1*24  1*-1 43  2*11  2*12  1* 2  1* 7  3*14  3*-1 41  2*11  2*15 1O* 2 LNS23
15*-1 4O 34*-1 -1 -1 -1 -1 -1 5O  1S 1E 1C 1U 1R 1I 1T 1Y 1  1I 1S 1  1A 1 LNS24
 1W 1A 1R 1M 1  1B 1L 1A 1N 1K 1E 1T-1                                    LNS25
-2                                                                        *EOP*
```

Sample Output

```
                    X X XXXXXXXXX
                  XXXXX          00
                 XXXX X    //      00
               XX X     //         00
              XXXX     //           00
              XX      //            00
             $$                      00
            $$$                       00
           $$$ $         ****   ****  00
           $$$ $         *****  *****  0
           $$$ $         *****  *****  00
          $$$$$$         *****  *****   00000
          $$$ $$         ****    ****        000
         $$$$ $$                                  00
         $$$$$$$                                    00
         $$$$$$                                      000
         $$$$$$$                                      000
         $$$$$$$                                       000
        $$$$$$ $$                                       000
        $$$$$$$ $$                     MMMMMMM          000
        $$$$$$$ $ 00                  MMMMMMMMMMM       000
        $$$$$$$ $  00                 MMMMMMMMMMM       000
       $$$$$$$$ $   0 000000             MMMMMMM        000
       $ $$$$$$ $    0     000000                       000
       $ $$$$$$ $    00       000                       000
       $$$$$$$  $     0       0  000              0000
       $    $$  $      0      0    0000000000000000
       $$   $ $        0      0    $ $  $$
        $$$ $ $       **********   $ $ $$$
          $$           0 ***** 00  $  $$$
          $$           00       00  $$ $ $
                      00        00   $$ $
                       00        00
                      00          00
                      00           00
                     00            00
                    00             00
                  000      0       00
                 000       0       000
                000  0     0        00
               000  0     00        00
              000   0     00        00
             00 000 0     00        00
            0000  000     00        00
           000     00     00       00
           000     00     00       000
          00       0      000     00 0000
          00       0    00000000 00000000000000
      IIIIIIOOO     0       0   0000000         00
  IIII IIIIIIOO     0                 000        00
III          0000000            0000    00   0   00          ------
IIIIIII   II        0              000   00   00  0   --------
     III   III      0         0      00   0    0  0
       IIIIII IIII  00         00      00  000  0 00          ------
                ----0000000000000     0 00 0000000  ----------
     ---------                00000000 00
```

LOVE IS A WET PUPPY DOG

```
                      DDDDDD   EEEEEEE   CCCCC   EEEEEEE  M     M  BBBBBB   EEEEEEE  RRRRRR
       999            D     D  E        C     C  E        MM   MM  B     B  E        R     R         666
      9   9           D     D  E        C        E        M M M M  B     B  E        R     R        6
   1    9999          D     D  EEEEE    C        EEEEE    M  M  M  BBBBBB   EEEEE    RRRRRR         6666   999
  11       9           D     D  E        C        E        M     M  B     B  E        R   R          6   6 9   9
   1    999            D     D  E        C     C  E        M     M  B     B  E        R    R          666   9999
   1                   DDDDDD   EEEEEEE   CCCCC   EEEEEEE  M     M  BBBBBB   EEEEEEE  R     R                  9
 11111                                                                                                      999
```

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
77777 7 7 7 7	1 11 1 1 11111 888 8 8 888 8 8 888	222 2 2 2 2 22222 999 9 9 9999 9 999	33333 3 33 3 3 333 1 000 11 0 0 1 0 0 1 0 0 11111 000	4 44 4 4 44444 4 1 1 11 11 1 1 1 1 11111 11111	55555 5 5555 5 5555 1 222 11 2 2 1 2 1 2 11111 22222	666 6 6666 6 6 666 1 33333 11 3 1 3 3 1 3 11111 333
1 4 11 44 1 4 4 1 44444 11111 4	1 55555 11 5 1 5555 1 5 11111 5555	1 666 11 6 1 6666 1 6 6 11111 666	1 77777 11 7 1 7 1 7 11111 7	1 888 11 8 8 1 888 1 8 8 11111 888	1 999 11 9 9 1 9999 1 9 11111 999	222 000 2 2 0 0 2 0 0 2 0 0 22222 000
222 1 2 2 11 2 1 2 1 22222 11111	222 222 2 2 2 2 2 2 2 2 22222 22222	222 33333 2 2 3 2 33 2 3 3 22222 333	222 4 2 2 44 2 4 4 2 44444 22222 4	222 55555 2 2 5 2 5555 2 5 22222 5555	222 666 2 2 6 2 6666 2 6 6 22222 666	222 77777 2 2 7 2 7 2 7 22222 7
222 888 2 2 8 8 2 888 2 8 8 22222 888	222 999 2 2 9 9 2 9999 2 9 22222 999	33333 000 3 0 0 33 0 0 3 3 0 0 333 000	33333 1 3 11 33 1 3 3 1 333 11111			

```
----------------------------------------------------------------------------------------------------------------------
```