11 11 TEX LATEX

HP48 Frequently Asked Questions List (FAQ)

Andre Schoorl <*aschoorl@engr.uvic.ca*> v4.60, 19 June 1999

Answers to Frequently Asked Questions about HP48 calculators

Contents

1 Introduction **6**

1.1 Overview . . . 6
1.2 This FAQ in Other Formats . . . 6
1.3 Translations of this FAQ . . . 7

2 Miscellaneous Questions **7**

2.1 I'm a novice, and I have some questions. Where should I start? . . . 7
2.2 What Usenet Newsgroups are there for the HP48? . . . 8
2.3 What are some good sites for the HP48? . . . 8
2.4 Are there any good WWW links for HP48 stuff? . . . 8
2.5 Is there an HP48 IRC Channel? . . . 9
2.6 How can I search for a specific HP48 file? . . . 9
2.7 What are good places to buy HP48's and accessories? . . . 9
2.8 What happened to EduCalc? . . . 10
2.9 Are there any CD-ROMs with HP48 software? . . . 11
2.10 Where can I find HP conference videotapes? . . . 12
2.11 Are there any HP48 Emulators? . . . 12
2.12 How can I make a dump of my HP48 ROM? . . . 13
2.13 Is the HP48 Allowed on SATs, AP, or ACT Tests? (U.S.) . . . 13
2.14 What is the HP49G? . . . 14
2.15 How does the HP48 compare with other calculators? . . . 17
2.16 What is the Universal Font Library (UFL)? . . . 21
2.17 What are the best alternate editors? . . . 22
2.18 Are there any PC fonts for the HP48? . . . 22
2.19 Is the HP48 serial number available in memory? . . . 22
2.20 What is the format of the HP48 serial number? . . . 22

3 Questions about ROM versions **23**

3.1 What does the SX/GX have that the S/G doesn't? . . . 23
3.2 What is the difference between the HP48 G/GX and the HP48 S/SX? . . . 24
3.3 What is the HP48 G+? . . . 24

4.6 What are all the different ON-KEY combinations? . . . 33
4.7 What are the different interactive self-tests? . . . 34
4.8 Why do I get an ESD error when I check my batteries with the self test? . . . 36
4.9 I can't account for some of the RAM in the HP48! . . . 36
4.10 What are the best methods for installing and removing libraries? . . . 37
4.11 How can I attach large libraries? . . . 38
4.12 What are those little numbers near the top of my screen? . . . 38
4.13 Can I use my HP48 as a remote control? . . . 38
4.14 Can the HP48 be used as a DTMF phone dialer? . . . 39
4.15 How can I run a program on warmstarts and/or power-up? . . . 39
4.16 I could speed up my HP-28S. How can I speed up my HP48? . . . 39
4.17 What is the hidden directory and how do I access it? . . . 40
4.18 How can I recall one object from a memory archive? . . . 40
4.19 Are there any key combinations that aren't in the manual? . . . 41
4.20 How do I find out about commands not in the GX manual? . . . 41
4.21 How can one delete (or purge) an entire directory? . . . 41
4.22 How does binary wordsize affect my calculations? . . . 41
4.23 Can I add my own equations to HP's Equation Library? . . . 42

5 Questions about Math, Plotting, and Equations 42

5.1 Why does (1/3)*3 equal 0.999999999999? . . . 42
5.2 Why does my HP say that 0 to the power of 0 equals 1? . . . 42
5.3 What is RPN? . . . 43
5.4 Why did HP use RPN in their calculators? . . . 43
5.5 Is there an alternative to calculating in RPN? . . . 44
5.6 Why do I get unexpected values when I add temperatures? . . . 44
5.7 How can I add my own units? What's the 1_? unit for? . . . 45
5.8 I want 'pi' to be a numeric value, not a symbol. . . . 46
5.9 What tricks are there for manipulating matrices? . . . 46
5.10 Can I make my HP graph any faster? . . . 46
5.11 How do I plot inequalities? . . . 46
5.12 How do I take the logarithm of a base other than 10 or e? . . . 47
5.13 How do I solve quadratic equations? . . . 47
5.14 Why do I get a complex answers for roots of negative numbers? . . . 48

5.15 Why doesn't '(-8)^(2/3)' give me the right answer (4)? . . . 48
5.16 How do I solve polynomials? . . . 49
5.17 How can I do statistics for grouped data? . . . 49
5.18 How do I generate a function table? . . . 51

6 Questions about cards, data transfers, batteries, and hardware 51

6.1 My HP48 seems to making a high pitched noise. . . . 51
6.2 Can I upgrade my S or G to more than 32K ram? . . . 52
6.3 Can I add a lithium battery backup? . . . 52
6.4 How can I get/build a cable for my HP48? . . . 53
6.5 HP seems to offer two link kits with different prices. . . . 53
6.6 Is there any communications software available? . . . 53
6.7 Why doesn't the I/R port work farther than a few inches? . . . 54
6.8 Can my HP48 talk to my HP100LX via Infrared? . . . 54
6.9 Can I use my HP48 to communicate with IrDA devices? . . . 54
6.10 Can I print to a LaserJet printer with an HP48? . . . 55
6.11 I downloaded a program but all I get is "HPHP48-..."! . . . 55
6.12 What is the structure of HP48 ASCII headers? . . . 56
6.13 Why do Kermit transfers seem to get slower? . . . 57
6.14 Why does XRECV not work sometimes? (GX) . . . 57
6.15 Additional Tips on Transfers . . . 58
6.16 What do the funny symbols in this document mean? . . . 58
6.17 What are the pinouts for the HP48 serial connector? . . . 58
6.18 Is there any information on interfacing to the HP48? . . . 59
6.19 How can I transfer programs/data from my HP-28S to my HP48? . . . 59
6.20 Can I use rechargeable batteries with the HP48? . . . 59
6.21 How can I tell, from within a program, if the battery is low? . . . 60
6.22 I lost the information on my RAM Card when I changed the battery! . . . 60
6.23 Why do I get an "Invalid card data" error when I merge a RAM card? . . . 60
6.24 I've heard about other manufacturer's RAM cards. Will they work? . . . 61
6.25 How does RAM card locking work? . . . 61
6.26 What are the advantages and disadvantages of covered ports? (GX) . . . 62
6.27 Why does the HP48 display flicker slightly? . . . 63
6.28 I broke the LCD screen - is there an easy way to get another? . . . 63

6.29 How can I protect my LCD? . . . 63
6.30 Is there a rigid (protective) case for the HP48? . . . 63
6.31 Can I use my Amateur Radio with my HP48? . . . 64

7 Questions about programs 64

7.1 Where can I get programs and information for the HP48? . . . 64
7.2 What are the Goodies Disks and where do I get them? . . . 64
7.3 How do I get access to the HP Calculator BBS? . . . 64
7.4 What are files that end with ".zip", ".Z", ".gz", or ".bz2"? . . . 65
7.5 What is a "ship" file? . . . 65
7.6 What is the ASC format and how can I use it? . . . 66
7.7 What is the HYDE library, and how do I get rid of it? . . . 66
7.8 What is the Minehunt game, and how do I use it? . . . 66

8 Questions about programming and advanced user functions 67

8.1 I've heard the names RPL, Saturn, STAR, GL etc... What are they? . . . 67
8.2 Is there a C compiler for the HP48? . . . 69
8.3 Why do SysRPL programs run faster than UserRPL programs? . . . 69
8.4 What is a good reference for learning SysRPL and ML? . . . 69
8.5 Can I make my own libraries? Can I split others? . . . 70
8.6 How do I know what library ID number to use for my program? . . . 70
8.7 What information is there on the internals of the HP48? . . . 70
8.8 Where can I get some programming development tools? . . . 71
8.9 I know UserRPL. How do I get started in SysRPL? . . . 71
8.10 Are there any viruses for the HP48? . . . 72
8.11 How do I store fields of variable length effectively? . . . 72
8.12 What is "Vectored Enter", and how do I use it? . . . 72
8.13 What is "WSLOG"? . . . 73
8.14 What are SYSEVALs? . . . 73
8.15 What are some useful SYSEVALs? . . . 73
8.16 What are LIBEVALs? . . . 75
8.17 What are some useful LIBEVALs? (GX) . . . 75
8.18 What is the format of a GROB object? . . . 75
8.19 What is the AUR and what information does it contain? . . . 77
8.20 What is the syntax for INFORM, CHOOSE, and MSGBOX? (GX) . . . 80

8.21 How do I put checkfields in my INFORM menus? . . . 83
8.22 What is the syntax for the INPUT command? . . . 83

9 Appendix A: Various Useful Functions 86
9.1 ASC Functions . . . 86
9.2 OBJFIX . . . 88
9.3 FIXIT . . . 88
9.4 LASTX . . . 90
9.5 Compact Data Storage . . . 91
9.6 HP82240B Printer Codes . . . 94

10 Appendix B: GX Specific Information 96
10.1 What's new in the HP48 G/GX? . . . 96
10.2 Examples of INFORM, CHOOSE, and MSGBOX . . . 103
10.3 Some useful LIBEVALs . . . 106

11 Appendix C: Details of Bugs 108
11.1 The EquationWriter Bug . . . 108
11.2 Rotation Rate to Angular Frequency Conversion Bug . . . 110

12 Appendix D: Hardware Additions 112
12.1 How to Make a Serial Cable . . . 112
12.2 Using a modem with the HP48 . . . 116
12.3 Additional Information on the HP48 and RS-232 . . . 119
12.4 Using Non-HP RAM Cards . . . 119
12.5 Where can one obtain third party RAM cards? . . . 122

13 Appendix E: Where to get HP48 Programs 122
13.1 Best Programs and Where to Get Them List . . . 122
13.2 Other Web / FTP Sites . . . 127
13.3 FTP by Electronic Mail . . . 128
13.4 Bulletin Boards with HP48 Programs . . . 128
13.5 HP Goodies Disks . . . 129

14 Contributors 130

1 Introduction

This FAQ is maintained by Andre Schoorl <*aschoorl@engr.uvic.ca*>

It was previously maintained by Keith Maddock, and was originally compiled by Darryl Okahata. Many thanks to both for their time and contributions!

I rely on your input in order to keep this FAQ up to date. If you have any suggestions or updates, feel free to mail them to me. If you have a correction or suggestion to make, please include the name of a specific question rather than its number as the number is automatically generated and can change between versions.

1.1 Overview

The purpose of this FAQ is to provide answers to commonly asked questions about the Hewlett Packard HP48 family of scientific calculators. This list contains information which has not necessarily been verified, and is not guaranteed to be correct. It has been compiled from various postings in the newsgroups comp.sys.hp48 and comp.sys.handhelds, as well as other sources. In particular, some parts were taken from older HP48 SX "FAQ Lists".

For some questions, there may be different answers for each calculator (G/GX vs. S/SX). In this case S/SX specific information is preceeded by an "SX:" on the left margin of the first line of each S/SX specific paragraph. G/GX information is denoted by a "GX:" in the same place. If a paragraph has no notation, then it is valid for all versions.

1.2 This FAQ in Other Formats

This document was made using the SGML-Tools (Standard Generalized Markup Language) package and is available in ASCII, HTML, and PostScript versions. All versions come from the same source, and are thus updated simultaneously.

The URL for the HTML version is `<http://www.engr.uvic.ca/aschoorl/faq/>`. The other formats are also available here in separate zip files.

The PostScript version is formatted for letter size paper, but it is also possible to create PostScript for legal size paper. Furthermore, DVI, LaTeX, Lyx, Info, and RTF forms of the FAQ are also possible. Since it is impractical to provide all of these formats, the SGML source is available in case anyone wants to create one of these formats on their own.

Text only versions of the FAQ will be posted to the newsgroup *comp.sys.hp48* as needed (usually every two weeks). Also, remember that all official FAQs (including this one) are mirrored at `<ftp://rtfm.mit.edu/>`.

Furthermore, it is posted to *comp.answers* and *news.answers*.

I will sign all text versions with PGP (Pretty Good Privacy) using the following key for authenticity, and provide MD5 Checksums for the remaining files. You can get a copy of this public key from the keyserver at `<http://pgpkeys.mit.edu/>`, or through the URL above.

```
Type Bits KeyID    Created    Expires    Algorithm       Use
sec+ 1024 2CFAA0BB 1997-06-21 ---------- DSS             Sign and Encrypt
sub  2048 F940E148 1997-06-21 ---------- Diffie-Hellman
uid  Andre P. Schoorl <aschoorl@engr.uvic.ca>
```

Alternatively, you can download the FAQ at one the following locations:

North America:

- <http://www.hpcalc.org/docs/faq/>

Europe:

- <ftp://ftp.stud.fh-heilbronn.de/pub/systems/hp48/incoming/>
- <ftp://ftp.stud.fh-heilbronn.de/pub/systems/hp48/info/FAQ/>

1.3 Translations of this FAQ

These translations may not be as up to date as the main FAQ, but should still be useful for international users who are more comfortable with their native language.

Portugese:

- <http://members.tripod.com/area48/> by Carlos Alberto Marangon

Spanish:

- <http://www.alumnos.utfsm.cl/aarrieta/hp48.html> by Alejandro Arrieta Rios and Guido Carvajal

2 Miscellaneous Questions

2.1 I'm a novice, and I have some questions. Where should I start?

Start by perusing the HP48 manuals. You'd be amazed how many questions can be answered if you read the manuals.

SX:

Earlier HP48 S/SX manuals came in two volumes. The first volume dealt with operating instructions, and the second volume dealt with programming. Later HP48 S/SX manuals combined these two volumes into one.

GX:

HP48 G/GX models come with a one-volume owners manual that covers all of the non-programming aspects of the calculator. An Advanced User's Reference (AUR) is available that covers programming aspects. It is highly recommended. They also come with a Quick Start Guide, a guided tour of some of the HP48's capabilities.

Also, look over the section "Answers to Common Questions", in Appendix A of the manual. If you have a two-volume manual, it will be in volume II.

2.2 What Usenet Newsgroups are there for the HP48?

comp.sys.hp48

Primary HP48 newsgroup.

comp.sys.handhelds

Occasionally, HP48 information is posted to this newsgroup.

comp.sources.hp48

This newsgroup has been dead for years, but was once a moderated group for HP48 Programs.

es.comp.sistemas.hp48

Is a spanish newsgroup for the HP48.

Articles posted to the newsgroup comp.sys.hp48 are archived at <ftp://wuarchive.wustl.edu/usenet/comp.sys.hp48/>. Articles are archived by year and month, with a general index covering approximately the past two years.

The AltaVista search engine at <http://www.altavista.digital.com> also accesses these archives (for a Usenet search), and automatically uncompresses any relevant articles you may select.

DejaNews <http://www.dejanews.com/> is another fast Usenet archiver/searcher, with many extra features such as Author profile and article search. The only drawback is that it strips attachments from posts (uuencode, MIME) to save space.

Another good way to find HP48 posts is through one of the various search engines available on the web. For example, Netscape, Lycos, and Magellan and others all seem to be able to find Usenet articles on the HP48. The search may even find articles that have long since expired from your local news server!

2.3 What are some good sites for the HP48?

Many HP48 programs can be found on the following FTP sites:

- <http://www.hpcalc.org/>
- <ftp://ftp.hp.com/pub/calculators/>
- <ftp://wuarchive.wustl.edu/systems/hp/hp48/>

For a list of other HP48 FTP sites, see Appendix E-2.

2.4 Are there any good WWW links for HP48 stuff?

Rather than list out a long list of HP48 related WWW resources, here is a single HP48 Page which contains links to other HP48 pages:

- <http://www.gmi.edu/madd0118/hp48/>

Another well maintained site is <http://www.hpcalc.org/>.

2.5 Is there an HP48 IRC Channel?

Yes, try connecting to an EFNet server and join channel #hp48. Many of the most knowledgeable HP48 users frequent it, making it a nice place to get your questions answered. There are even two IRC robots (hp48b and RPL) on the channel. The following information will get you started:

First, download an appropriate IRC client. Collection of clients are available at <http://www.irchelp.org> or <ftp://ftp.undernet.org/>

Second, find a working EFNet server and join #hp48. A list of servers is given at <http://www.irchelp.org/irchelp/networks/servers/efnet.html>

2.6 How can I search for a specific HP48 file?

If you know the all or part of the filename of an HP48 program, you can use the proper Internet search services instead of bothering all the readers of the newsgroup. The following server is especially good, it seems to find HP48 files from just about anywhere.

<http://ftpsearch.ntnu.no/ftpsearch>

2.7 What are good places to buy HP48's and accessories?

X-Philes CD-ROM:

There is a CD-ROM called the X-Philes on the market that includes many HP48 files. See the next question for more details.

Office Max:

Toll Free: (800) 788-8080

Wholesale Products:

This is another company on the web that sells HP handhelds, including the HP48. Their address is <http://www.wholesaleproducts.com> or you can mail inquiries to <*wholesal@shore.net*>

The Mall At Wholesale Products
400 West Cummings Park
Suite 1725-122
Woburn, MA 01801
781-438-8622 (Telephone, 9 a.m. - 5 p.m. Boston time)
781-438-8307 (FAX, 24 hours a day, 7 days a week, from anywhere)
E-mail: wholesal@shore.net

Web:

North America:

- <http://www.wholesaleadvantage.com/hpcalcs.html>
- <http://www.waterw.com/jake-s/handyclc.htm>
- <http://www.dvtg.com/>
- <http://www.smi.com/>

- <http://www.tdsway.com/>
- <http://www.ramss.com/>
- <http://www.chotkeh.com/>
- <http://muffet.com/samsoncables/>
- <http://www.calculatorsinc.com/>

Europe:

- <http://members.aol.com/weidatec/>
- <http://www.Digitalis.de/>
- <http://stolte-edv.com/>
- <http://home.t-online.de/home/FFFFF/>
- <http://www.bbm.e.se/>
- <http://www.geocities.com/Eureka/Enterprises/3190/>

Others:

```
Global Connections Incorporated (formerly Connection Point, Inc.)
50 South River Street, Suite 105, Janesville, WI 53545
Tel: (608)-752-9548 or (608)-752-1537

Oxford Educational Supplies (U.K.)
Tel: 01869 343369 (mail order)
Notes: No shipping charge for within U.K.

Spectrum Office Equipment
22 Stokes Croft Bristol (U.K.).
0117 9428278

Best (Western U.S., used to be La Belles) 1-800-950-2378

Calculators Inc., Minneapolis, MN, U.S. (612) 866-8908

Carrington, (800) 982-3731

J&R Music World - Computer World 1-800-221-8180

Office Depot (U.S. and Canada)

Service Merchandise (East Coast U.S.)
```

2.8 What happened to EduCalc?

EduCalc, which had been one of the best sources for HP48's and accessories, went out of business on 31 December 1997.

2.9 Are there any CD-ROMs with HP48 software?

Yes, there are a few CD-ROMs with HP48 software I know of:

HP48 Archive CD:

The most current set of files at <http://www.hpcalc.org/> is always available on a custom burnt CD-R. Details are at the same URL.

X-Philes:

The X-Philes is a CD-ROM made by Synchron Data that contains most HP48 programs to date. All together the CD-ROM contains over 21,000 files and more than 620 MB of information, mostly zipped.

The HP48 sections contain 3194 files and make up 38.5 MB of the total. This includes all of the Goodies disks (aka Horn disks) 1 through 10.

There are also files for HP95, Psion, TI-82 and TI-85 calculators, as well as information and programs on:

Jokes, Star Trek, Sports, Midi, History, Strange files, Conspiracy, Law and Business, Internet & Networks, Atari, BBS-Systems, Hardware, Computers, 3D-Objects, Anarchy, Hacking, Movies, X-Files, Music, Magazines, Science, UFO and Aliens, Occult, Security & Privacy, Amiga, Cyberspace, Subcultures, Radio, Virus hunting, Survival, Phreaking, PC utilities, Game cheats and levels, and Programming.

The CD comes with easy to use Windows software for browsing and searching for keywords. This software contains an upload feature that allows you to transfer programs directly to your HP (or modem) using X-Modem.

More information on obtaining the X-Philes CD-ROM is at:

WWW: <http://www.algonet.se/synchron/>

E-Mail: <*dt93tn@pt.hk-r.se*>

Regular Post:

```
Synchron Data
Tranbarsvagen 25:14
372 38 Ronneby
Sweden
```

SHAREWARE Volume 1 for Hewlett-Packard:

There is a CD-ROM by a German company called Yellow Computing named SHAREWARE Volume 1 for Hewlett-Packard. It sells for DEM 49.80 (U.S.$ 34.00), and is distributed from:

```
Yellow Computing                Phone +49 7136 951143
Computersysteme GmbH              Fax +49 7136 951111
Postfach 1136
D-74173 Bad Friedrichshall
Germany
```

This is the German company which offers PC <-> HP-48 transfer cables together with Windows software for transfering files (the "Transfile Win 48" package).

The CD-ROM comes with a 16 page booklet in German and English which shows how to use the included Windows 3.1 transfer program (with program and catalogue files in both languages).

It's sampled in Aug 95, pressed Oct 95 and contains about 85 MB of software for the HP48 SX/GX, 95LX, 100LX and 200LX; the Horn disks 1 through 9, and about 710 additional HP-48 programs in the 9 categories: Animation, Data, Games, Graphic, Programming, Maths, Memory, Science, Utility and Communications. It also has about 200 other unsorted, compressed HP48 files. Decompression software is included.

It's not an ISO CD-ROM, since it contains directories and files, whose names contain exclamation marks, hyphens and dollar signs.

Handheld HP CD-ROM Library:

This collection contains 13,611 files for a total of 314 MB of compressed software for HP Handhelds. The principal groups of this collection are:

1. HP28C/S (up to 2.5 Mb) Applications, Games, Utils, Docs.
2. HP48S/SX (up to 48 Mb) Applications, Business, Chemistry, Communications, Emulators, Games, Graphics, Math, Physics, Programming, Sound, Time, Utils, Docs.
3. HP48G/GX (up to 40 Mb) Applications, Business, Chemistry, Communications, Games, Graphics, Math, Physics, Programming, Sound, Time, Utils, Docs, Sys-RPL.
4. HP38G (up to 2 Mb) Applets, Applet Development Kit, Docs.

Platform files total 30 MB, additional documentation is 18 MB, various collections including the Horn disks and SysRPL collection total 106 MB, and finally 58 MB of information from HP groups around the world.

For more information, orders, and pricing send e-mail to *<carles@idgrup.ibernet.com>*

Ingenieur Professional CD ROM

This is another German CD issued by Harald Kresin and Franz Lorenz. It contains roughly 67 MB of HP Software. Their web page is `<http://home.t-online.de/home/05713201317>` and their e-mail is *<05713201317@t-online.de;*

Their address is

```
Franz Lorenz
Pocketcomputer & Zubehoer
Bruchstr. 51, D-32423 Minden Germany
```

2.10 Where can I find HP conference videotapes?

Jake Schwartz maintains a list of videotapes of HP handhelds meetings / demos / conferences going back as far as 1986 and the intro of the HP18C. They are available at relatively low cost to anyone who is interested. The URL is `<http://www.waterw.com/jake-s/video.htm>`.

2.11 Are there any HP48 Emulators?

A fully functional HP48 emulator by Eddie Dost called "x48" is available. It runs in X-Windows on any Unix based operating system like Linux or Solaris. However, it requires a ROM dump of an HP48 to run. Of course, due to copyright laws such a ROM dump cannot be distributed with the x48 package.

Therefore, you must be the proud owner of an HP48 (S/SX or G/GX) in order to create your own ROM dump. The emulator will look and act either like a S/SX or a G/GX, depending on what type of ROM dump you give it. Further instructions are contained in the x48 package itself. The latest version of x48 is available at <http://www.hpcalc.org/pc/emulators/>

There is also a freeware emulator for Windows called Emu48. It is written by Sebastien Carlier, and runs under either Windows 95 or NT. It offers many exciting features, including fast speed, saving of memory, direct loading of files onto the stack, a greatly customizable interface, and support for ports through 33. It supports both S/SX and G/GX ROM dumps, and comes with a "RomDump Wizard" to help you create one.

The binaries and full C++ source code, distributed under the GNU Public License, as well as more information, are available at <http://www.epita.fr:8000/sebc/Emu48/>. There are also a series of "Unofficial Service Packs" available at <http://privat.swol.de/ChristophGiesselink/>.

An emulator for the HP38, based on Emu48 has also been created. It is available at <http://www.epita.fr/avenar_j/hp/38.html>

There is also an HP48 S/SX/G/GX Emulator for Amiga computers called Alcuin. See <http://www.cis.tu-graz.ac.at/home/schupfer/Alcuin/index.html> for more information.

2.12 How can I make a dump of my HP48 ROM?

Emulators for the HP48 generally require an image of your HP48's internal ROM, also known as a "ROM Dump". Currently the best program to make a dump of your ROM is ROMDump Wizard by Christoph Giesselink. It is available at <http://privat.swol.de/ChristophGiesselink/>

2.13 Is the HP48 Allowed on SATs, AP, or ACT Tests? (U.S.)

From: Jack Levy

The S/SX/G/GX are allowed on the SAT's. The S/SX/G/GX are allowed on the SAT II Math IIc test, but no calculators are allowed on any of the others. The S/SX/G/GX are allowed on the AP Calculus exam. However, only the S/SX are allowed on the AP Chem and AP Physics exams because the G/GX has the built in equation library.

Update from: Sandler Rubin

As of 1996, HP 48G calculators are allowed on the physics and chemistry APs, but only for the free response section. All calculators are now banned on the multiple choice sections on the physics and chemistry APs. As for calculus, the HP 48G is allowed on one half of the multiple choice and all of the free response. The other half of the multiple choice on the calculus test must be completed without any calculator.

These changes have occurred because the AP tests are now more conceptual and depend less on pure "number-crunching". In the free response sections of the chemistry and physics tests, lists of important physics equations are include for student use which basically make the HP 48G EqLib redundant.

Update from: John Goerzen

After a lot of discussion with Candance Noble <*Noble@act.org*> of ACT, they changed their policy to allow it only if the IR port is covered with opaque tape. Their response follows:

This notice is in response to questions ACT has received about acceptable calculators. Please pass this clarification on to others.

Effective Fall 1996, examinees may use calculators on the ACT Assessment and PLAN Mathematics Tests. Students taking the ACT Assessment should review page 3 of Registering for the ACT Assessment. Students taking PLAN should review page 4 of the Planning Guide for Students and Parents. All makes and models of calculators are acceptable – including programmable calculators and hinged calculators with mathematical formulas printed by the manufacturer on the inside flap – provided the calculator does not have any of the unacceptable features specified in the ACT publications noted above and further clarified below. Examinees using unacceptable calculators will be dismissed from testing.

The following information is provided to help students select an acceptable calculator if they choose to use one on the ACT Assessment or PLAN Mathematics Test.

POCKET ORGANIZERS, HANDHELD OR LAPTOP COMPUTERS

All such devices are prohibited.

ELECTRONIC WRITING PADS/PEN INPUT DEVICES

Such models are readily recognizable due to the pen or stylus used to input data. All such devices are prohibited.

"QWERTY" KEYPAD

The term QWERTY derives from the first six letter keys (left to right) of the top row of typewriters, personal computers, most handheld and laptop computers, pocket organizers, and specialized calculators. All models with QWERTY keyboards are prohibited.

PAPER TAPE

If a calculator has a feature that provides paper output, the paper must be removed to make the calculator acceptable.

NOISE

Calculators that have a "talking" or audio feature are acceptable in a group testing situation only if the sound feature has been turned off.

WIRELESS COMMUNICATION

Calculators that can communicate (transfer data or information) wirelessly with other calculators can be recognized by the "window" or infrared data port on the top edge of the calculator, similar to one on a television remote control. Such calculators are acceptable only if the wireless transfer capability is disabled by placing opaque material (such as masking tape) over the infrared data port.

POWER CORD

If a calculator can run on either battery or electrical cord, the electrical cord must be removed to make the calculator acceptable.

2.14 What is the HP49G?

On May 21st, 1999, at the OpenHP Conference in Paris, France, Hewlett Packard announced the HP49G graphic calculator. The following information comes from Jean-Yves Avenard of HP's ACO (Australian Calculator Operation).

Hardware

- 512KB of RAM
- 2MB of flash memory (1MB used for upgradable ROM, 1MB available to user)
- 4MHz Saturn CPU
- 131x64 pixel screen (black instead of blue, so higher contrast)
- 51-key keyboard with tactile rubber keys
  - This is not the same type of keyboard as the HP48 has but it is still an HP quality keyboard. Rubber keys don't always mean bad quality, as there are a lot of mechanical parts to make a keyboard.
  - The feeling is extremely similar to the HP48 keyboard. Do you really think that HP could let a new product go out of its doors with a keyboard that will break after 6 months?
- New keyboard layout - a user friendly keyboard that incorporates the strengths of the existing RPN keyboard layout and the more familiar algebraic style keyboard layouts on the HP 38G, TI-83 and TI-89.
- Metallic blue case with a translucent blue-tint slide-on cover. You can put it in the HP48's case if you prefer a soft case.
- No IR, but an HP49<->HP49 cable is provided (adapter included for connection to HP48). There were problems in some countries with some teachers fearing the HP48 would be used to cheat because of the IR port.
- RS-232 serial port with Kermit (binary, ASCII) and Xmodem (128 checksum, 1K, 1K CRC) running at 9600 bps (15360 bps internally, but no PCs support that speed).
- Any unit may be directly connected to a data-logger, overhead projector, a personal computer or another HP 48G Series or HP 49G graphing calculator.
- The hardware is not expandable, but the ROM can be software-upgraded and there is plenty of memory, so this should not be a problem.

Memory Management

- Even though the Saturn can only access 512K of memory, a new bank switching routine is used, in addition to absolute addresses, making it much faster.
- The user sees three ports of memory: Internal System RAM (256KB, port 0), Extended RAM (256KB, port 1), and Flash User ROM (1MB, port 2).
- The system will manage how to copy your files for you, so there is no need to get lost in a bank.

Software

- Step-by-step solving option that enables you to learn how to get the right answer numerically or symbolically.
- The most complete built-in Computer Algebra System (CAS) currently available on a calculator for fast, advanced symbolic manipulation and solving.
  - Imagine an HP48 with Erable and ALG48 plus everything that is missing. Now speed it up so it can compete (and be faster) than anything else on the calculator market. That's the HP49's CAS.

- Inferential and Descriptive Statistics (suitable for Advanced Placement and College/University level statistics).
- A fast, flexible and intelligent Editor for isolating, editing, manipulating and evaluating text, equations, expressions, sub-expressions, programs and graphs.
- Input and Output Customization - choose between "textbook" form (pretty print), algebraic and RPN input and "textbook" and algebraic output in various font styles and sizes.
  - When you first boot the machine, it's an algebraic machine. If you don't like algebraic mode, simply press Mode, select "RPN" and that's it. You're back to the HP48 Reverse Polish Notation mode forever and ever (at least until you clear the memory or manually switch back to algebraic). It works EXACTLY like the HP48, but faster.
- Between 10 and 100 times faster than the existing HP 48G Series
  - The OS has been completely rewritten, mainly in assembly language, so nearly everything that was slow on the HP48 works faster, including input forms, choose boxes, stack display, command line, file manager, etc. When I say faster, I mean MUCH faster. The HP49 is even faster than an HP48 with the MetaKernel running in it.
- There is no longer a built-in equation library but the constants library remains.
- Do you think the 4MHz CPU is slow? The 3D real-time plotter can draw 6 frames (for a 14*14 points matrix) per second... It's faster than any current competitors! You can rotate in real time over the X axis, Y axis and even the Z axis.

Programming

- Provides four programming languages:
  1. HP Basic: like a hybrid between RPL and Basic. This does not have GOTO and cannot really be called Basic, as it looks like the HP38G's programming language. Example code:

```
FOR(i,1,100)
    DISP(i,1);
    IF I+1-5==50 THEN DISP("Hello World",2) ELSE DISP("I'm off",2) END
STEP(1)
```

  2. User RPL: Just like the HP48's built-in language. To reuse a User RPL program from an HP48, send it to the 49 using ASCII Kermit.
  3. System RPL: Development software, including disassembler, built into ROM but not supported.
  4. Assembly language: Development software, including disassembler, built into ROM but not supported
- Grayscale support (in ASM as well as System RPL). All System RPL graphic routines work on grayscale graphics.
- Vectorized Interrupt System (add your own interrupt handler)
- To run an HP48 application on the HP49, just recompile it. It will probably run faster, too, especially if it uses the GUI.
- Entry points have moved to make the ROM easier to maintain, but programs will still be compatible at the source code level.

The HP49 is the most powerful calculator ever announced at any time. It works in two ways: one fully compatible with the HP48 series, including RPN but faster, and the other algebraic, easy for users of other calculator brands to learn to use.

Although you may not agree with some of the specifications above, please wait until you touch it before complaining. After all, the HP49 was developed by HP48 users and tested by HP48 users!

For more information, visit the HP Calculator Archive at <http://www.hpcalc.org/>

2.15 How does the HP48 compare with other calculators?

From: Chin-Yu Hsu <*chinyu@voyager.bxscience.edu*>

What are the differences between the HP48 and the TI?

Before listing the differences, I wish to let you know that the HP48 has much more functionality than the TI's. However, these additional functions are usually of no help in high school than the TI. Instead, they are only helpful in college depending on the course you take. The TI is probably a better choice on tests in high school since it is easier to use (to most people) and does calculations faster (because it offers speed at the expense of the higher precision of the HP48). Please note that the HP also offers the HP38 which is more comparable to a TI-82.

Also, you should realize that if you want support for the graphing calculator, the HP48 is the one to get. Although you can get support from Texas Instrument (by phone and by web), and also by teachers in school, you can get support for the HP48 from a larger group in the world. There are far more web sites related to the HP48 than to the TI graphing calculators. Also, the HP48 has its own newsgroup and its own IRC channel (usually up 24 hours a day and maintained by a bot that can send you programs at your request). Please note that although the calculation and graphing capabilities may not be available for a particular calculator, you can usually download a program that extends such a capability to it.

If you find that part of the information below is incorrect, please tell me about it. You may need to provide additional proof, if I am unable to confirm it. Any additional information on the TI-92 is helpful because much about it is still a mystery.

General Differences:

Calculator	HP38	HP48(2)	TI-82(3)	TI-85	TI-92
List Price(4)	$109	$135/$265	$125	$135	$250
Available Mem.(5)	32KB	32/128KB	28.2KB	28.2KB	70.0KB
Batteries Req.(6)	3AAA	3AAA	4AAA+1	4AAA+1	4AA+1
Plug-in Cards	No	Yes(7)	No	No	Yes (Plus)
Infrared Transmit.	Yes	Yes(8)	No	No	No
Wire Transmit.	Yes	Yes	Yes	Yes	Yes
Cal-to-PC trans(9)	Yes	Yes	Yes	Yes	Yes

Overhead display	Yes	Yes(10)	Yes	Yes	Yes
CBL support(11)	No	No*	Yes	Yes	Yes
QWERTY keys(12)	No	No*	No	No	Yes
Screen capture	Yes	Yes	via link	via link	via link
Screen Resolution	131x64	131x64	96x62	128x64	240x128
Character Display	22x8	22x8	16x8	21x8	?????

Miscellaneous Functions:

Calculator	HP38	HP48(2)	TI-82(3)	TI-85	TI-92
Simple Eq. Solver	Yes	Yes	No	Yes	Yes
Simult. Eq. Solver	?????	Yes	No	Yes	Yes
Poly. Root Finder	No limit	No limit	No	30th order	Yes
# of matrices	10	No limit	5	No limit	No limit
max matrix size(12)	No limit	125x125	15x15	30x30	99x999

Graphing Functions:

Calculator	HP38	HP48(2)	TI-82(3)	TI-85	TI-92
Simultaneous Graph	Up to 10	No limit	Up to 10	Up to 99	Up to 99
Zoom features	14	15	13	15	?????
Graph Tracing	Yes	Yes	Yes	Yes	Yes
Fnc. Value Tables	Yes	No	Yes	No	Yes
Parametric Graphs	Yes	Yes	Yes	Yes	Yes
Polar Graphs	Yes	Yes	Yes	Yes	Yes
Recur. Seq. Graphs	Yes	Yes	Yes	Yes	Yes
Diff. Graphs	No	Yes	No	No	No
Conic Graphs	No	Yes	No	No	No
3-D graphs	No	7 types	No	No	Yes
Area under Curve	No	Yes	Yes	Yes	Yes

Extremum	Yes	Yes	Yes	Yes	Yes
Roots	Yes	Yes	Yes	Yes	Yes
Slopes	Yes	Yes	Yes	Yes	Yes
Tangent Lines	No	Yes	No	No	???

Statistical Functions:

Calculator	HP38	HP48(2)	TI-82(3)	TI-85	TI-92
Scatter Graphs	Yes	Yes	Yes	Yes	Yes
X-Y line Graphs	No	No	No	No	Yes
Box-Whisker Plots	Yes	No	Yes	No	Yes
Histogram Plots	Yes	Yes	Yes	Yes	Yes
Regression Graphs	No	Yes	Yes	Yes	Yes
Bar Graphs	Yes	Yes	Yes	Yes	Yes
Truth Graphs	Yes	Yes	No	No	No

Notes:

1. In the specifications, "no limit" usually means that it is limited by memory only.
2. There are five different HP48 calculators. There are the HP48S, HP48SX, HP48G, HP48G+, and HP48GX. The S and SX models are older models, and no one would be getting them anymore since the G, G+ and then GX offer all the same features, but with additional enhancements to make the features easier to use. The G, G+, and GX are also 40% faster than the S and SX models. The difference between the G and the GX is that the G has 32K of RAM, and no expandability, while the GX has 128K of RAM and the ability to have two additional plug in cards. The G+ is in between, having 128K of RAM but no expandability.
3. There is a new version of the TI-82, called the TI-83 which should be available soon. It offers all the features of the TI-82, except it has additional statistics and financial functions, as well as displaying a table and a graph side by side. Please note that this feature is not available on the HP48, but is available on the HP38. Also, the TI-83 has an equation solver like the HPs.
4. Cost for HP calculators does not include cables because they can transfer data via infrared. The price for TI's include the cables for linking one TI to another. Also, the HP48G's list price is $135.00, while the HP48GX's list price is $265.00.
5. The available memory listed for the HPs are slightly lower than the given values. The available memory for the HP48G is only 32 K. The HP48GX has 128K of available RAM, and is expandable to 4.5 megs!

6. The batteries used by these calculators are AAA batteries. Rechargeable batteries are not recommended because they have a much lower life span than alkaline batteries. The TI's require an additional CR1616 battery to retain their memory power. The HP's do not need this additional battery, but once you take out the AAA batteries, you have about 3 minutes to replace them with new ones before it loses its memory. The RAM cards for the HP48GX requires an CR1616 battery, however, if you plan to store it outside of the HP.

7. Only the HP48GX support plug-in cards. The HP48G doesn't.

8. If you are thinking of getting the HP's to cheat on tests because of the infrared capability, forget about it. The infrared receivers on the HP's only have a maximum range of THREE inches (The HP's transmitter has a longer range, up to several feet, but that will not help at all unless you are using it as a remote control to watch TV, or you are sending to an infrared device connected to an AC power source, such as the HP LaserJet 5 printers).

9. The Calculator to PC (or MAC) requires additional cable and software. For the HP, you only need to get the cable, which can be made from standard computer parts (a DB-9 and CD cable totaling about $15). No additional software is needed because you can use any communications program of your choice. For the TI, you can either need to purchase a proprietary software and cable for about $60 to $80 at stores or you can download the software free at TI's web site, and try to make your own cable. The same type of package is available from HP, but offers programming tools and debuggers, and costs about the same price.

10. The overhead display unit for overhead projectors is available only for the HP48GX because the HP48G does not have the available extension slot to install the necessary interface card.

11. The CBL stands for Computer Based Laboratory. It is created by TI, so it is not available on the HP of course. It simply allows you to collect data from different sources (sounds, temperature) at set intervals and can be analyzed on the calculator. The CBL unit's list price is approximately $225. Please note that there is a similar device for the HP-48, called the ADCM-48 which is a A-to-D converter and I/O interface that can read from four analog and four digital inputs.

12. Please note that calculators with QWERTY keys are not allowed on the SAT exams. The HP calculators do not have QWERTY type keyboards, but you can connect it to your computer and have the keys from your computer echo on to the HP calculator after installing a simple freeware program such as those found in `<http://www.engr.uvic.ca/aschoorl/archive/as-comm.zip>`.

13. The maximum matrix size on the HP48G is only 42x42. The maximum matrix size on the HP48GX is 85x85, but it can be expanded to 125x125 with the addition of an extra RAM card.

14. The TI-92 manual states lists can be up to 999 elements, and data can be arranged in 99 columns of 999 elements each.

From: Bernard Parisse

- Overhead display: there are special models of the TI92 to do this, standard models give bad results
- Memory: The TI92-II model has 198K memory (price in France 1890 FF instead of 1390 FF)
- Processor: Modified 68010 at 10MHz

- Cabri-Geometre is very slow on the TI, the teachers I met at my lecture told me that they never use Cabri on the TI.
- Numeric computations: The HP clearly beats the TI, especially in linear algebra (factorization of matrices, eigenvalues/eigenvectors, etc...)
- Symbolic computations: The TI clearly beats the HP... without addition of any math programs. Hence I compared the TI adding the best programs available on the net (i.e. ntheory (10K) for number theory, odeproj for ordinary differential equations (40K), that was all since I had only a 70K TI92) with my HP loaded with Alg48 and Erable. The main advantage of the TI92 is speed (about 5-10 times faster than Erable, 1-2 times faster than Alg48). The main drawback on the HP48 side is the lack of a limit instruction. Some features of Alg48 and Erable are not provided by the TI92:
  - Factorization of integers (limited to factors less than or equal to 65520 on the TI92, integers may not have more than 614 digits on the TI92) and of polynomials (e.g. the expanded form of:

    ```
    (x^4+x^3+1)*(x^4+x+1)
    ```

    is not factored on the TI92).
  - Integration: the Risch algorithm is not implemented in the TI, e.g. it can not find the antiderivative of

    ```
    (1+2x^2)*exp(x^2)
    ```
  - Taylor series: This feature is probably implemented like in the HP48 ROM, hence the TI can not solve series(sin(x)/(exp(x)-1)) at x=0 for order greater than 4.
  - Eigenvalues/Eigenvectors/Jordanisation of matrices
  - Systems of polynomials equations (Groebner bases)
  - Ordinary differential equations are only solved with programs like odeproj, no support for Laplace or inverse Laplace transform
  - A lot of internal programs are not accessible to the end-user on the TI92, as the source code of Alg48 and Erable is freely available.
- Programming: Assembler is now available on the TI, but there is no counterpart to SysRPL and the build-in language is comparable to a shell-script langage (with some limitations e.g. for usage of programs in functions)

2.16 What is the Universal Font Library (UFL)?

The Universal Font Library, or UFL for short, is a set of libraries designed to help reduce the overhead of replacement fonts between a number of major HP48 programs. The basic idea is instead of each program containing its own replacement font(s), the fonts are stored in a cetral library to save room. In addition, the UFL comes with a Font Maintenance Library (FMnt).

If you use some of the programs in the Best Programs List, you may find you need a particular UFL library installed in order to use the program. The UFL is maintained by Andre Schoorl and is distributed under the GNU General Public License. For more information and to obtain the UFL package, see `<http://www.engr.uvic.ca/aschoorl/ufl/>`.

2.17 What are the best alternate editors?

If you like a smaller font, I would recommend TED by Mika Heiskanen, or MiniWriter by Jean-Yves Avenard. Both use FNT1 from the Universal Font Library (UFL) to allow font customization and to save space.

MiniWriter is very small and fast, and has cut, copy, paste, and search features. It is available at http://www.epita.fr/~avenar_j/

TED has slightly more features, such as a character map, but as a result is larger. TED is based on ED, which is the editor that comes with the Jazz library. The latest version is available at <http://www.engr.uvic.ca/aschoorl/>.

If you prefer the medium font, I would recommend HP-Writer by Paquot Christophe. It's quite small, fast, and uses FNT2 from the UFL. Its features include cut, copy, and paste, as well as a macro key and built in User RPL key support. However, it only works on G/GX. The current version is available at <http://www.geocities.com/SiliconValley/Lab/1930/>.

Other editors include StringWriter, also by Jean-Yves Avenard, which is a very complete and full featured library. EDEN, by HPFox, is another popular editor in France. Another is QED, which was originally written by Lutz Vieweg but is now maintained by Rick Grevelle.

2.18 Are there any PC fonts for the HP48?

Yes, there are several fonts on the Goodies Disks (see the Appendix), as well as some TrueType fonts available at <http://www.hpcalc.org/pc/misc/>

2.19 Is the HP48 serial number available in memory?

No. There is no electronic version of the HP48 serial number. This was apparently considered by the design team, but it was determined that the costs would greatly outweigh the benefits.

2.20 What is the format of the HP48 serial number?

The HP48 serial number is printed on the back of the HP48, just above the battery compartment. There are two different serial number formats. The older format was used when the HP48 was first introduced and continued through the first part of 1997. The newer format has been used on HP palmtops since about 1993 and was started on the HP calculators in the first part of 1997.

The new serial numbers are in this format:

```
ccywwnnnnn
```

cc

Place of manufacture (new two-letter code)

The known codes are:

- SG = Singapore
- ID = Indonesia

- MY = Malaysia

y

Year of manufacture. This represents the last digit of the year. A 7 would mean 1997.

ww

Week of manufacture (01-53).

nnnnn

The unit number for that week. (first unit on Monday is 00001)

For example, if your serial number were `SG72706543`, the unit was the 6543rd unit manufactured in Singapore during the 27th week of 1997.

The old serial numbers are in this format:

```
yywwMnnnnn
```

yy

The year the unit was made (add 1960 to this value)

ww

Week of manufacture (01-53).

M

The manufacturing location (A for America, S for Singapore)

nnnnn

The unit number for that week. (first unit on Monday is 00001)

For example, if your serial number were `3321S07509`, the unit was the 7,509th unit manufactured in Singapore during the 21st week of 1993.

3 Questions about ROM versions

3.1 What does the SX/GX have that the S/G doesn't?

SX:

The HP48 S is the same as the HP48 SX except that the HP48 S does not have the two expansion slots of the HP48 SX. This means that the HP48 S cannot be expanded and can only access 32K of memory, versus 288K for the HP48 SX (two 128K RAM cards). (Actually, there are third-party RAM cards that contain 256K or 512K on a single card – this memory is accessed via a bank-switching mechanism.)

GX:

The HP48 G is the same as the GX except for two differences. The GX has the 2 expansion ports, and also has 128K RAM built in, whereas the G only has 32K.

If you plan on heavily using your calculator, you will probably find that 32K is not enough memory. In fact, some of the best HP48 Programs will not even fit in this memory!

3.2 What is the difference between the HP48 G/GX and the HP48 S/SX?

The HP48 G/GX calculators are, basically, slightly faster (by 40%) versions of the HP48 S and HP48 SX calculators, with more features, such as 3D-plotting and a new forms-based interface. Much of the HP Solve Equation Library card is now built-in. See Appendix B-1 for a detailed list of GX features.

Note, however, that the HP48 G/GX is different from the HP48 S/SX in a number of significant areas:

1. The HP48 GX comes with 128KB RAM, as opposed to 32KB RAM in the HP48 SX. The HP48 G still has 32KB RAM, however. The HP48 G/GX also has twice the ROM (512K), compared the HP48 S/SX (256K).
2. Much of the HP48 G/GX internals have changed, compared with the HP48 S/SX. Unfortunately, this means that many programs written in assembly language or System RPL won't work on the HP48 G/GX. This is especially true of any program that tried to access the display memory directly.
3. The HP48 G/GX CPU runs at 4 MHz, instead of the 2 MHz used with the HP48 S/SX CPUs. However, due to various overheads (memory bank switching, etc.), the speed increase between the S/SX and G/GX is less than 2 times. G/GX throughput is approximately 40% faster.
4. Slot 2 of the HP48 GX can access up to 32 ports (ports 2-33) by bank switching. Each port is 128KB, for a maximum total of 4 MB in slot 2.

3.3 What is the HP48 G+?

The HP48 G+ was announced 30 March 1998. It is essentially a HP48 G with more memory - it has 128K of memory instead of 32K. All G+ machines have revision R ROMs.

There is no change to other aspects of the calculator model offerings. Specifically, it looks like HP has added to the HP48 family without withdrawing anything. The G, G+, and GX will all continue to be sold.

The press release for the HP48 G+ is at: `<http://www.hp.com/pressrel/mar98/30mar98.htm>`

3.4 How can I tell what ROM revision I have?

GX:

Type the command "VERSION". Note that this command is not part of any key or menu, and so you must spell it out.

Type #30794h SYSEVAL. It returns a string, "HPHP48-x", where x is your ROM revision. This works for both SX and GX.

SX:

1. Turn the calculator on (press ON and then release).
2. Press ON again, and hold it down.
3. While holding down the ON key, press and hold down the "D" key (the fourth white key in the top row).
4. While holding down the "D" key, release the ON key.

5. Release the "D" key. The display should be blank, with the exception of three vertical lines (one down the centre, and one down each of the right and left sides).
6. Press the backspace key (it says DROP and CLR over it). A meaningless string of digits should appear in the top line of the display (mine says "705D9:1B8DA178E5A111B6" – yours may be different).
7. Press and hold down the EVAL key. It should say something like the following, where the ? is an uppercase letter indicating the ROM version:

   ```
   Version HP48-?
   Copyright HP 1989
   ```

8. Release the EVAL key. Hold down ON and hit "C" (the third white key in the top row. This is a general reset operation that returns the calculator to normal from the diagnostic stuff (which is where ON-D puts you).

3.5 What bugs exist in the various versions of the HP48?

Rom: A

- DEFINE rounds numbers in user-defined functions if STD is not the current display mode.
- KGET on a zero length file from another machine will cause the calculator memory to be cleared.
- NXEQ from the Graphics Environment with flag -3 set (SYM off in Modes) can overwrite the current equation with a number.
- Displaying the clock in 24-hour format causes the one-digits for minutes to be incorrect in the alarm catalog.

Roms: A-C

- INV returns an incorrect result if used on an 8x8 or larger matrix. To get around this problem, divide the identity matrix with the matrix to invert.

Roms: A-D

- Entering a symbolic complex number in polar form like (A,<B) (where < is really the funny little angle symbol), gives an expression in A and B that is wrong (the A and B are interchanged so A is the angle and B is the magnitude).
- If the clock is displayed during an ARCHIVE via RS-232, there is a chance (not 100%) that calculator memory will be cleared, after the transfer. The work-around is to turn the clock display off before doing the transfer. Should your memory be cleared, you will have to restore the contents of memory from the archive.
- TRNC and RND allow for an array in level 2 and a symbolic in level 1 of the stack. This allows you to put an array in a symbolic. Example: [2 3] '7/8' RND returns 'RND(UNKNOWN,7/8'

Roms: A-E

- There's a bug in the EquationWriter that is "fixed" in the Rev J ROM. See Appendix C-1 for details.

Roms: E, J

- BESTFIT: Occurs when a two dimensional array is stored in \GSDAT (SigmaDAT). Pressing the BESTFIT softkey multiple times will leave a lot of useless reals on the stack.

Rom: K

- First HP48 G/GX released; includes various bugs affecting ports. Therefore, the version K ROM was only used in HP48 G units.

Roms: K, L, M, P

- XROOT Function fails when used with list processing unless X=Y. It does not corrupt memory.
  FIX: Preceed any XROOT which will use list processing with a SWAP

Rom: L

- Second HP48 G/GX released; fixed the port bugs. This ROM version is that present in the 2,000 HP48 GXs sold at the ASEE conference for $90 apiece.

Roms: L, M

- HP48 GX may lose all of user memory when: clock is displayed, a library object is be stored in user memory, and a RAM card is lugged in to port one and merged with User Memory. If all four of these conditions are present and the calculator is turned on when the clock is being updated by the system, a loss of all User Memory may occur.
  FIX: Just type the following in on your calculator:

```
<< RCLF -> f << -40 CF OFF f STOF >> >> 91.3 ASN

Then turn on USER mode
```

- EquationWriter/Character Picker Bug: If the size of the equation is greater than the regular size of the screen, calling the Character Picker in Equation Writer will garble the screen.
- If you store an object into a port > 1 while there is a card installed in slot 1, the end-of-obj-sequence marker (00000) is written to port 1 instead to the destination port. This will cause the 'Invalid Card Data' warning to pop up each power on after a while and, in addition, the memory of port 1 gets currupted (if it's writeable). A temporary fix is to remove the card from slot 1 when changing the contents of the card in slot 2. However a better fix is to get the STOFIX library (available on GD#10 or at HPCVBBS, ~1K in size).
- Tail Bug: The TAIL command returns an error on single length strings. This was fixed to return an empty string in later Roms. Examples:

```
"X" TAIL --> Error: Invalid Dimension (Rom M)

"X" TAIL --> "" (Rom R)
```

Rom: M

- Evaluating a tagged object which contains an operator (+, -, *, /, etc) as the second or subsequent character will leave an External and System Binary on the stack, as well as the original object.

Roms: L, M, P

- SigmaDAT: Put a NON-MATRIX in \GSDAT (SigmaDAT), run STAT graphical apps, edit \GSDAT. Causes a crash.
- XRECV will fail or cause memory lost if the amount of free RAM left is not about twice the size of the expected file.

  FIX: See FXRECV on GD 9. Note that FXRECV is not necessary for Rom R, and in fact will not even run properly on Rom R.
- RREF does not transform the matrix into Row Reduced Echelon Form correctly if the first column of the matrix consists entirely of zeros. It may also return incorrect answers in some matrices.

  A work-around for this bug is to delete any leading all-zero column(s) from the matrix; this may be done using the Matrix Writer, via its -COL menu key, or via commands: 1 COL- DROP (to drop the first column). Since row-reduction operations never change any all-zero column, you already know in advance that the given column(s) must be all-zero in the final result.

Roms: L, M, P, R

- portnum: { dir ... var } RCL/EVAL fails if portnum >= 2
- Rotation rate to angular frequency conversion bug. See Appendix C-2 for a complete write-up.

Roms: M, P, R

- Repeated executions of FFT or IFFT can cause a memory loss. The FFT and IFFT warmstart problem is caused by a CPU return stack overflow during interrupt handling. The problem can only occur when the argument to FFT or IFFT is an MxN matrix with both M > 1 and N > 1. The vector argument case is immune to this problem.

  FIX: To compute the FFT of a matrix without risking warmstart, expand the matrix into its row vectors using ->ROW, compute the FFT of each row vector, collect the results into a matrix using ROW->, expand the matrix into its column vectors using ->COL, compute the FFT of each column vector, and finally collect the results into a matrix using COL->. Except for less precision, this is the same result as would be obtained by applying FFT on the original matrix. Use the same procedure with FFT replaced by IFFT to compute the IFFT of a matrix.

  Here are program examples implementing the "Matrix FFT" and "Matrix IFFT":

```
MFFT     << ->ROW FFTLP ROW-> ->COL FFTLP COL-> >>
MIFFT    << ->ROW IFFTLP ROW-> ->COL IFFTLP COL-> >>
FFTLP    << -> n << 1 n START n ROLL FFT NEXT n >> >>
IFFTLP   << -> n << 1 n START n ROLL IFFT NEXT n >> >>
```

- Equation Writer/Equation Library interaction bug: If you pick an equation from the EQ library, enter solver for that equation, enter an algebraic on the stack like 'X^2', and press the down arrow key to view algebraic in EquationWriter, the name of the equation will appear in the EquationWriter along with the unassociated algebraic.

Roms: M, P

- BESTFIT: Occurs when a two dimensional array is stored in \GSDAT (SigmaDAT). Pressing the BESTFIT softkey multiple times quickly can cause a warmstart, or SigmaDAT to be erased. This occurs sometimes just by pressing BESTFIT repeatedly, regardless of \GSDAT content. This bug can also be triggered by pressing the BESTFIT softkey quickly after pressing any other key. Simply pressing MODL then BESTKEY quickly will trigger the bug.

Rom: P

- PROOT: If you put a matrix of coefficients on the stack and press PROOT repeatedly, you will get a warmstart.

Roms: A-R

- MatrixWriter Bug: This shows up when deactivating and reactivating any of the two direction movement fields. If you do this, the active matrix field won't move as it's supposed to. For example: Launch the matrix writer (check that your GOFORWARD field is activated). Type [1] [ENTER]. The active matrix field moves forward as it should. Deactivate the GOFORWARD field and activate it again. Now enter [2] [ENTER] [3] [ENTER]. The digits should show up on the same row, but they don't, because after you deactivated and reactivated the GOFORWARD field, the active matrix field stops moving according to the movement fields.
- DTAG Bug: When the `DTAG` command is executed without any arguments on the stack, the error message "XLIB 0 0 Error" is given, which is non-standard.

Roms: K-R

- Equation Library Solver and Multiple Equation Solver Bug: Pressing MSOLVR or MUSER after setting user flag 63 can cause Memory Clear. Example: CLEAR, 63 CF, EQNLIB, down-arrow, ENTER, down-arrow, [SOLV], NXT, 63 SF, right-shift MUSER, DROP –> Memory Clear. This bug is also found in the HP Solve Equation Library Card (HP 82211B, but not 82211A) for the SX.

3.6 What is new with the revision "J" ROM? (S/SX)

The revision J ROM has a faster equation writer, as well as some bug fixes. Scrolling of graphics is also faster.

One "problem" is that the revision J ROM changed some undocumented ROM entry points, which broke those programs that used them. If you have a Rev J ROM, you will not be able to run some old programs.

3.7 What happened to ROM revisions G, H, I, O, and Q?

These versions either were never released to the public, or were skipped over.

I have received confirmation of the existence of a HP48 S with ROM revision F. So it appears revision F was in fact released to the public, although very few must have made before revision J came out.

3.8 Can I upgrade my HP48 S/SX?

When the G/GX first came out, HP had a trade in program. This is no longer in operation. The only option remaining is to try to sell your S/SX privately.

3.9 Can I upgrade my buggy GX to the latest ROM revision?

As far as I know, not any more. They had a program a in early '94, but it is over now. Contact the following for more info:

```
Hewlett-Packard Corporation
Calculator Support
1000 NE Circle Blvd
Corvallis, OR 97330

Phone: (970)-392-1001
```

3.10 What's the latest ROM version?

- Revision R is the latest ROM version of the G/GX series.
- Revision J is the last ROM version of the S/SX series.

3.11 Why would I buy a S/SX rather than a G/GX?

Here's the top reasons to buy a S/SX instead of a G/GX (Thanks to all who contributed):

1. May be slightly cheaper, having only 32K RAM.
2. It isn't THAT outdated.
3. So what if it's a little slower?
4. You didn't need all those useless equations, graphics capabilities, and math and science functions anyways!
5. If you break it, you're out of less money!
6. You like seeing how much you can fit into 32K RAM.
7. You like slow file transfers, using Kermit instead of X-Modem.
8. You want to be able to use all those lists of SYSEVALs for the SX.
9. You like having only 3 ports available max. 33 is ridiculous!
10. Since it doesn't have a built in equations library it is less likely to be banned on tests.
11. Only wimps need GUI's, dialog boxes, and choose lists.

4 Questions about the Operating System and Using the HP

4.1 How do I find the checksum of an object?

Many documents (including this FAQ) make references to the "checksum" of an object. This is a 16-bit user binary resulting from a CRC calculation on the contents of an object. This binary is supposed to be relatively unique, with only 1 change in 65536 of accidental equal checksums on two different objects. This allows you to distinguish programs that look the same but may be quite different (even if the size of them is the same). It is also often used to verify correct transmission of files.

You can find the checksum of an object using the BYTES command. This will return you two things - the size of the object in bytes on level 1, and the checksum on level two. Note that while the checksum of a variable name is the same as running the checksum on the object itself, the sizes will be differ by 4.5 bytes + the size of the variable name itself.

In rare cases the checksum of two objects can be the same, even if the objects are different. This is due to the limited nature of the HP48's checksum function. You can use the SAME command in these cases.

4.2 I typed RULES on my G/GX and got a crossword puzzle!

From: Joe Horn

Many commercial programs (like Windows) have similar hidden screens that list the program's design team. Since they are always so cute and well hidden, they are often referred to as "Easter eggs". The RULES command on the G/GX is one such example.

Try using the RULES command. The names you see are as follows:

```
/----------------------\
|            M G       |
|      P   DIANA       |
|    CHARLIE X B     D |
| BILL U   N  TED    A |
| O  A L   N    ALCUIN |
| B  I    JIM   V      |
|  RON     S    E      |
\----------------------/
```

ALCUIN was the HP-internal code name of the HP48 G during development, because Alcuin was Charlemagne's teacher, and Charlemagne, as you remember, was the code name of the HP48 SX.

The other names in the "Easter egg" above belong to:

- BILL Wickes (list processing; "Father of RPL")
- CHARLIE Patton (RPL operating system)
- TED Beers (parameterized outer loop; interactive stack; key handling system; high-level display management; input forms)
- DIANA Byrne (project manager; plotting; graphics)

- GABE Eisenstein (EquationWriter)
- BOB Worsley (I/O)
- PAUL McClellan (unit management; math)
- CLAIN Anderson (product manager; marketing)
- DENNIS York (manager)
- JIM Donnelly (EQ LIB; list processing; variable tic logic; developer support)
- MAX Jones (menu system; editing)
- DAVE Arnett (hardware)
- DAN Coffin (manuals)
- RON Brooks (marketing)

4.3 Why does my HP48 occasionally "freeze" for a moment?

The HP48 must occasionally do "garbage collection" to free up unused memory, and it is this garbage collection that is causing the momentary "freeze". "Garbage collection" is where the HP48 scans through memory, looking for objects that are no longer used.

Using less stack supposedly makes garbage collection go faster.

The easiest way to force a garbage collection is to run MEM DROP. Otherwise you can use #05F42h SYSEVAL or the SysRPL command GARBAGE.

4.4 What do I do if my HP48 locks up?

- NOTE: If you have it, read the sections in the GX manual starting on page 5-16, "Special Memory Operations", and "Testing Calculator Operation" on page A-9.
- NOTE: under most circumstances, your calculator should never lock up. Some earlier versions of the G/GX do this due to bugs. Of course, depending on the program being run, it may appear to lock up, but pressing the "ON" key usually interrupts whatever the calculator is doing. If it does lock up, this is usually caused by a buggy assembly-language or system RPL program, and, chances are, the memory contents of your HP48 are gone (as in "destroyed", "terminated", "wiped out", "deleted", etc.).

1. First, check the batteries to make sure that they are installed correctly.
2. Next, try is pressing the "ON" key again and again very RAPIDLY. The operative word here is "rapidly". It's possible to write a program that can only be interrupted during a very narrow window, and so you must press the "OK" key again and again very rapidly to interrupt the program. If the calculator doesn't respond within ten or so seconds, go on to the next step.
3. Press "ON-+" a couple of times, just in case the LCD display intensity was turned down. Here, "ON-+" means:
    - Press and HOLD DOWN the "ON" key.

- Press and release the "+" key, WHILE HOLDING DOWN THE "ON" KEY.
- Release the "ON" key.

4. If this doesn't work, the next thing to try is ON-C. This is a "warm-start" or "system halt". Note that this will CLEAR the stack. Here, "ON-C" means:
   - Press and HOLD DOWN the "ON" key.
   - Press and release the "C" key, WHILE HOLDING DOWN THE "ON" KEY.
   - Release the "ON" key.
5. If this still does not work, it is possible that the HP is not responding because it is not accepting input from the keyboard. It is possible to execute a system halt without the keyboard using the hidden reset button. The reset button is hidden under one of the two upper rubber feet on the bottom of the calculator (it's under one of the two feet near the serial connector and I/R port). Note that the location of the reset button (under which foot) varies from calculator to calculator. If you gently pull out the rubber feet, you'll notice that there is a small hole under each foot. The reset button is located in the hole that has an "R" next to it. Once you've located the correct hole, you press the button by GENTLY sticking an unbent paper clip, into the hole with the "R" next to it. Hold for one second and remove.
6. The next thing to try is a complete memory reset. Depending on the state of your calculator, this may or may not preserve the contents of memory. At the very least, even if the contents are preserved, the contents of the HOME directory will probably be moved into a new directory below HOME (HOME will be empty with the exception of this new directory); you'll have to manually move the items back into HOME. To perform a complete memory reset, press ON-A-F. This means:
   - Press and HOLD DOWN the "ON" key.
   - Press and HOLD DOWN THE "A" key, WHILE HOLDING DOWN THE "ON" KEY.
   - Press and HOLD DOWN THE "F" key, WHILE HOLDING DOWN THE "ON" AND "A" KEYS.
   - Release all of the keys.
   - At this point, the calculator will ask you if you want to recover memory. Answer "yes" to this question. Note that this can take a long time. If you answer "no", all of memory will be cleared (the old contents will be lost).
7. If that fails, you should still be able to induce a complete memory reset by removing your batteries and pressing ON several dozen times. This will drain the capacitor that stores information when the batteries are removed, and you should get a "Try to Recover Memory?" prompt if you have pressed ON enough times to reset your calculator. If this doesn't work you can try taking the batteries out for a few hours and then putting them back again. You should definitely try this method before even considering the next step!
8. **WARNING**: the next method is HIGHLY unrecommended! Permanent damage to your HP can result if you try this. In particular, you could fry a couple diodes used to protect your HP48. You have been warned!

   If your HP48 does not respond to any of the above attempts, you might consider reversing the polarity of 2 of the 3 batteries **VERY BRIEFLY**, pressing ON, then correcting the batteries. Your HP48 should start with a "Try to Recover Memory?" prompt. Even reversing only 2 of the 3 batteries is dangerous, but it is safer than reversing all 3 batteries.

Note that simply shorting the battery terminals will not discharge the storage capacitors. There are diodes between the battery terminal and the storage capacitors which prevent the DC current from flowing backward.

This method apparently drains the capacitors and reverse charges them by running current through the parasitic input protection diodes in the various CMOS chips.

9. If all of the above fail, or if you decide not to try the last step, you may have to contact Hewlett Packard for service.

4.5 My HP48 seems to take longer to turn on and off.

The usual causes for this are RAM/ROM cards and libraries. When you turn the calculator on, it checks RAM (the more you have, the longer it takes), and checks to see if any libraries need initializing.

Also, having a lot of alarms can supposedly also cause this.

4.6 What are all the different ON-KEY combinations?

From: Dave Arnett

ON

In most cases, this will move you to, or toward, the stack environment.

ON+A+F

This is the manual Coldstart. It will break out of almost any hung program, and offer you the dreaded option... "Try to recover memory?" Note that this is your next-to-last resort in a lock-up, so don't give your machine this three-fingered salute unless you really want to possibly clear memory.

ON+B

This is the Oops! key for ON codes. For many ON sequences, this cancels the operation, so long as you press the B before releasing the ON key. Try it sometime, if you must, just so you remember it. It can be a life-saver, in case of an accidental ON+A+F in process, or even accidentally hitting ON during a plotting routine.

ON+C

This will initiate a Warmstart. You'll lose the stack and PICT, but it will often get you out of a locked up program with memory intact.

ON+D

Enters interactive test mode. Back up your own machine and play around. Exit this test mode with Warmstart, ON+C. See the next question for all the various tests available here.

ON+E

Starts a looping self-test mode. Kinda dull after the first five minutes, but it keeps a rather solid tempo. Exit with Warmstart, ON+C.

ON+4

Cancels repeating alarms.

ON+1

Screen dump utility.

ON+'+' and ON+'-'

Adjusts display contrast.

ON+SPC

Initiates Coma Mode and clears Warm Start Log. In Coma mode, all the clocked activity is halted. The battery power drain is basically the few nano-amps [nA] necessary to sustain your RAM contents. The Warm Start Log is a nearly bullet-proof area of memory which keeps track of the last four hazardous events. View it using the command WSLOG. This log is one of the few things not cleared by a Coldstart. But it is cleared by ON+SPC. Enter Coma this way if you want to clear the Warm Start Log, if you are planning to study crystal healing in Tibet for a few years and want to keep your pirated version of Tetris alive, or you just have a low power fetish. Exit Coma mode with the ON key (tough to remember, huh?), and plan to find your stack cleared out.

Two additional notes on this Coma stuff. First, I'm not gonna try to list the log codes in WSLOG. Sorry! Second, if your machine is on, and you drop the batteries out, you will usually end up in a coma mode to preserve power. WSLOG will not be cleared. Instead, you will find a code 1 entry there. Some people are paranoid and want to be in Coma when they change batteries, just in case they have a sudden emergency call from the Prime Minister and don't get back to their battery change for a half hour. If you are one of these... shall I say, weenies? No, that would be unprofessional ...users, then I recommend you use the ON+SPC entry route, rather than the kamikaze method of dropping batteries with the machine running. Myself, I just turn the machine OFF, like the manual says.

Now, a Coma story... sort of. During the G/GX development, I maintained a small number of units on which I changed ROM chips as incremental code releases came out. This was so folks like Bill Wickes and Jim Donnelly could have fully-real hardware to test, instead of just the EPROM handsets some of you may have seen. We always backed up anything of importance in the machine before we did this. But I got into the habit of using ON+SPC before I opened up the calculator. Better than half of the time, I could remove the batteries, open the case, desolder the old surface-mounted ROM chip, solder down a new chip, and reassemble the machine... with all of RAM intact! When I pressed ON, I'd come right back up without a "Try to Recover Memory?" prompt. Pretty good, huh? The folks who put that mode into the machine certainly weren't in a coma.

4.7 What are the different interactive self-tests?

Pressing ON-D enters the interactive self test function of the HP48. When you first press ON-D, your HP will beep and you will see 3 vertical lines, one on each side of the screen and one in the middle. You can exit this mode with a Warmstart, ON-C. The tests are:

[A]

Displays CPU speed

[B]

LCD test. Press Enter to cycle through the tests

[C]

Internal ROM test

[D]

Internal RAM test

[E]

Keyboard test. Starting with [A], press all the keys in order, left to right, top to bottom. If all goes well, the HP will show "KBD1 OK".

[F]

Partial keyboard test

[G]

ESD test monitor. Battery status is shown by bars.

[H]

UART loop-back test

[I]

Wired UART echo

[J]

Shows what cards are plugged in. Press any key a few times.

[K]

Test RAM cards in ports

[L]

Blanks display

[M]

Sends system time from IR port

[N]

Receives system time from IR port (have another HP send it)

[O]

IR loop-back test

[P]

IR UART echo

[S]

Shows test start time

[T]

Shows test fail time

[U V W X Y Z]

Looping test

[ENTER]

Initialize test times

[DEL]

Test summary

[BACKSPACE]

Enters memory scanner (SX only)

When the HP48 is in test mode (ON-D or ON-E), the test results that are displayed on the screen are also sent to the serial port at 9600 baud, 8 bits, no parity, 1 stop bit. This seems to be unaffected by any settings in IOPAR, PRTPAR, or the system flags. I can only guess that this feature was included for factory testing purposes. This does explain, however, why the I/O annunciator lights up so much during the self test.

4.8 Why do I get an ESD error when I check my batteries with the self test?

From: Detlef Mueller

If your HP48 batteries are low, try this test with fresh alkaline AAAs. This test first records the CRC of a RAM segment, then starts a loop where it creates power peaks and recalculates the CRC again until you press a key or the CRCs differ. In the latter case one or more cells of your RAM were "Electro-Statically Discharged" and the test exits with the last CRC done. If there's a hardwaer problem, this number should differ in consecutive executed ESD tests - with fresh AAA alkalines! Old ones or non alkaline batteries may cause trouble under this test condition.

If the test runs continuously, the bars on the screen reflect the battery status (low bat if bar is there) of port 2 (below the 2), port 1 (below the 1), and the main AAAs (below 0). If the AAAs are very low, the V bar comes up.

4.9 I can't account for some of the RAM in the HP48!

The HP48 S/SX/G have 32K base memory, and the GX has 128K. However not all this memory is available to the user. The operating systems uses memory for system variables, flags, fonts, last stacks, last arguments, last menus, and such. It also can be any one of a number of things:

1. Enabling the use of LAST STACK, LAST ARG, LAST CMD, etc., uses up extra RAM, as the calculator has to keep track of the last stack, arguments, command, etc.. This is particularly important with LAST STACK if you have a large/complex stack, as the HP48 has to keep enough information around to recreate the stack after each operation. You can clear out these variables by pressing ON-C (but also note that this will clear out the stack, the graphics PICT, etc.)

2. From William Wickes, a member of the HP48 development team: Here's a candidate for "most obscure way to use up memory on the HP48 SX." When a program executes DOERR with a string argument, the string and hence the program too if the string is embedded in the program are referenced for the sake of the ERRM command. Thus if you purge the program, the memory used by the program is not reclaimed until the ERRM reference is updated to a new string. If the string is in a port, you will get the Object In Use error if you try to purge the library or backup object containing the string. A system halt prior to a PURGE will not help unless the object is in port 1 or port 2 (it will reclaim

memory held by a successfully purged but still referenced object). For port 0 objects, the only way to break the reference is to execute DOERR again with a new string.

3. Occasionally, I have had problems with libraries disappearing, but still taking up memory. I found that an "ON-A-F" reset will cause these libraries to re-appear. If you still cannot account for a large chunk of memory, I would suggest a hardware reset, after backing up your memory.

4.10 What are the best methods for installing and removing libraries?

To install a Library:

1. Download it into a variable of the calc.
2. Place the library on the stack.
3. Type the number of the port in which you want to store the library in. 0 is appropriate if you do not have any expansion cards.
4. Press STO to store the library in that port.
5. Turn the calculator off and then on, or press ON-C, to finish installation of the library.

Sometimes you have to purge the variable containing the library before storing it in a port, if you are low on memory. See the next question for more information.

If you have FXRECV installed: Just type a port number, hit FXRECV and the library will automatically go into the specified port. All you have to do then in turn the calc off and then on to complete the installation.

To delete a Library:

Say you have library 1234 in port 0 that you want to delete.

1. Go to the directory to which the library is attached. In most cases this will be HOME.
2. Enter :0: 1234
3. Press [ENTER] or type DUP
4. Type DETACH
5. Type PURGE

If this doesn't work, there may be a reference to the library still in temporary memory. Try warmstarting with ON-C, then try again.

Libraries with very low library numbers may require the addition of a constant in order to delete them. For example, to delete a library 4 in port 0 you can add 2048 to the library number, as follows:

1. Type :0: 2052 DETACH
2. Type :0: 4 PURGE

Therefore, here is a fairly general program for detaching and purging most auto-attaching libraries; the only required argument being the real library number (e.g. 218 in the above case):

```
\<< PATH SWAP HOME DUP 2048 + DETACH DTAG "&" \->TAG PURGE EVAL \>>
```

4.11 How can I attach large libraries?

If have just successfully transferred a library to your HP, and try to store it in a port you may get an "Insufficient Memory" error. This is because your HP is trying to make a duplicate copy to store in your port, but it doesn't have any memory to do so. This is especially true on the HP48 S/SX/G since they only have 32K of memory built in.

The solution is to push the library onto the stack, then purge the original variable containing the library. At this point, the only copy of the library that exists is on the stack. Now you can store the library in a port as usual. This method avoids temporarily trying to have two identical copies in memory at the same time.

For example, say you have a library in 'LIB' and you wish to store it in port 0. You would enter the following:

- Go to the directory containing 'LIB'
- Recall the library to the stack: **'LIB' RCL**
- Purge the original from user memory: **'LIB' PURGE**
- Enter port number to store in: **0**
- Press **STO**

For the opposite problem - sending large libraries you can follow the reverse procedure. That is, recall the object from the port to the stack, purge the original in the port, store it in a variable and then send it.

Alternatively if the library is in a non-covered port (0/1/2 on SX or 0/1 on GX) you can use the following simple program suggested by Dan Kirkland:

```
\<< -> lib \<< 'lib' SEND \>> \>>
```

This does not require a second copy of the library and avoids all the complications of trying to purge a library and then re-installing it.

4.12 What are those little numbers near the top of my screen?

The little numbers are "user flags". When you or a program sets any of the user flags 1 through 5, little numbers corresponding to the set flags are displayed at the top of the screen.

The HP uses negative flags for setting various modes and functions, while positive flags were reserved for the use of the user. To set a user flag type the number, then SF. To clear a user flag type the number, then CF.

Some programs will use the user flags without you being aware of it. So, if you have a "1" at the top of the screen you can type 1 CF to get rid of it.

4.13 Can I use my HP48 as a remote control?

There are numerous programs available for using your HP48's IR port as a remote control. However, the level of support depends on the particular unit you wish to control. The HP48 is capable of reproducing many of the IR controls of many popular brands of appliances.

However, some remotes run at speeds that the HP48 is unable to reproduce. Due to their faster processor speed, the G/GX calculators are able to support more remote controls but are still limited in some cases.

With typical remote programs you have to "teach" your HP48 by sending it the various IR sequences with the original remote control. There are some advantages to using your HP48 however:

- You can impress your friends that you can use your calculator to control your TV!
- You can set alarms to turn your things on and off at predetermined times.
- You can use it as a universal remote.

The best remote program by far is Remote by Bjorn Gahm. It has a large database of existing remotes, meaning you may not have to "teach" your HP48 your specific remote if it is already supported. However it also have the ability to learn new codes. See the best programs list (later in the FAQ) for the specific URL.

4.14 Can the HP48 be used as a DTMF phone dialer?

To make phone dialing tones, you need to produce two simultaneous tones. The tones used to dial phones are known as DTMF (dual tone multi frequency) tones, which consist of two tones at the same time.

The speaker in the HP48 is not capable of supporting the DTMF tones because it can only make a single tone at a time. The best the HP48 can do is switch very quickly between one frequency and another. Some attempts have been made in machine language to support DTMF, however they have been unsuccessful. Therefore, you cannot use the HP48 as a phone dialer.

4.15 How can I run a program on warmstarts and/or power-up?

You can use some of the utilities available on the Goodies Disks (see Appendix E.5 on where to obtain these):

- AUTOBOOT (GD #7) warmstarts every power-up
- AUTOEXEC (GD #10) Runs 'GO' every power-up
- AUTOPORT (GD #7) Runs &:AUTOEXEC every warmstart
- AUTOSTRT (GD #7) Runs AUTOEXEC every warmstart

Some stack libraries come with support for AUTOEXEC or an equivalent built in (for example, the Java library supports AUTOEXEC). If one of the libraries you have installed supports such a feature, you may not need any of the utilities above.

4.16 I could speed up my HP-28S. How can I speed up my HP48?

You can't, really. Unlike the HP-28S, the clock speed in the HP48 SX is hardwired to 2 MHz, and to 4MHz in the HP48 GX.

Digitalis `<http://www.Digitalis.de/>` has a "speed-up" module available. They sell 48G's with this module added (it looks like it adds about 50DM/$30 or so to the cost). I'm not sure what it does, but I

think it boosts the clock speed. Apparently it can be turned on and off as needed, at the sacrifice of shorter battery life.

However, in some cases, you can speed up the HP48 by turning off the display refresh, which supposedly takes up about 11% of the CPU time. Detlef Mueller submitted a program called SPEED to comp.sources.hp48 that turns off the display refresh. Of course, since it turns the display off, it is only suitable for long non-interactive runs like calculating large integrals, compiling, etc... Look in the various HP48 archives for a copy.

4.17 What is the hidden directory and how do I access it?

The "hidden directory" is a subdirectory of the HOME directory. It is created on startup, and it contains variables with your alarms and user key configurations. You can use the hidden directory to store whatever you want, however you CAN NOT ORDER the contents of this directory, as this causes memory loss.

The name of the hidden directory is a null character, ''. The only way to get a null character is to use a syseval. You can use the following program to access the hidden directory:

```
<<  HOME                          (changes to home dir)
    #15777h              (syseval to get null character)
    SYSEVAL                       (returns '' to stack)
    EVAL >>                     (changes to hidden dir)
```

To exit the hidden directory, use HOME or UPDIR. This will return you to the HOME directory.

Also, by putting the null character as a part of your HOME custom directory, you can switch right into the directory by pressing the softkey.

Note: The SysRPL command NULLID returns a null character to the stack; it is the same as doing #15777h SYSEVAL.

4.18 How can I recall one object from a memory archive?

From: Joe Horn

There is a little-known method of recalling objects from within archives. Just make a list representing the PATH to the object, and tag the list with the port number. Then hit RCL or EVAL.

For example, if your archive in port 1 is called BKUP, and it contains a directory called GAMES which contains a program called REVERSI, you can run the game directly (without having a separate copy of REVERSI stored anywhere!) by:

```
:1: { BKUP GAMES REVERSI } EVAL
```

Or, you can RCL instead of EVAL, and it will be pushed onto the stack.

- This method only works in ports 0 and 1 in a GX. It works on any port in the SX.
- This method works with directory objects stored as backup objects, as well as with the results of the ARCHIVE command, because they are both really the same thing.

4.19 Are there any key combinations that aren't in the manual?

From: John Meyers

Yes, on the G/GX there are a few keys that aren't mentionned. They are:

- [right-shift] [down-arrow] = Review menu keys
- [right-shift] [right-arrow] = Kermit server mode

[Right-shift] [View] is hidden in Appendix G (Operation Index) of the User's Guide [Edition 4] (which comes with the calc), among the several entries for the word VIEW.

AUR [Edition 1] (in chapter 3, Command Reference, under SERVER) gives the alternate keys [right-shift] [cursor right] (but of course AUR doesn't come with the calc). This alternative was left out of Appendix G of the User's Guide. Donnelly's HP48 Handbook [2nd Ed.] even lists it first!

4.20 How do I find out about commands not in the GX manual?

From: John Meyers

Besides purchasing the Advanced User's Reference Manual (AUR), you can look in Appendix G of the G/GX User's Guide, which briefly summarizes every command available in the G/GX, including all the commands not mentioned anywhere else in the User's Guide. Often even the brief description given in this Appendix is sufficient for you to use the command (and to know it exists!).

Some notes on omissions, however:

- VERSION and "_" (attach unit) are missing in Edition 4.
- SUB works with arrays in the G/GX, but this is not in the HP manuals, nor in Donnelly's HP48 Handbook 2nd Ed 6/93. You can specify any two "corners" of a rectangular sub-matrix (like with GET), and it extracts the sub-matrix. Also works with vectors.

4.21 How can one delete (or purge) an entire directory?

- To purge a subdirectory and all that it contains, type in the name in single quotes, then PGDIR.
- To purge all variables in the current directory, but not the directory itself, type CLVAR.

Note that with CLVAR, if the current directory contains a non-empty subdirectory, an error occurs at that point, and no subsequent variables are purged.

4.22 How does binary wordsize affect my calculations?

From: John Meyers

Some of the following (but not all) is in the User's Guide, under "Binary Arithmetic and Number Bases" (Chapter 15 of G/GX).

The system flags contain a segment which specifies a "binary word size". This word size limits how many low-order bits of a binary value are displayed on the stack, and equally determines how many low-order bits of the arguments and results of binary arithmetic/logical/shift operations are used.

Curiously, the binary word size has no effect on entering values into the command line - what you originally enter is displayed in truncated form on the stack, but when you view it again in the command line (which temporarily reverts back to STD numeric display and 64-bit wordsize) you will find the complete value as entered is still intact.

As soon as you begin using functions on that value, however, the arguments to binary functions get truncated to the current wordsize before the functions are executed, and then the final result of the function gets truncated again.

You can recall the current wordsize using RCWS and set the wordsize using STWS. STWS accepts a real number argument in the range 1-64 (out of range values get adjusted to either 1 or 64), and it also accepts a binary argument, in which case the absolute value of the low-order 20 bits is used (just in case you wondered why #1048576d STWS RCWS gives an answer of 1).

Don't try adjusting the binary wordsize in the hope of saving memory by using shorter values; every "user binary integer" (type 10) produced by User RPL occupies 13 bytes, no matter what the current wordsize; the HP48 just sets the leading bits to zero when displaying on the stack or performing calculations.

4.23 Can I add my own equations to HP's Equation Library?

No, the Equation Library is in ROM, and can't be written to. There are several programs which allow you to create your own equation library such as EQBrowser, EQ Lib, Equation Manager, and xMGR.LIB. All of these programs can be found at `<http://www.hpcalc.org/utils/misc/>`.

5 Questions about Math, Plotting, and Equations

5.1 Why does (1/3)*3 equal 0.999999999999?

This is due to the way numbers are typically represented by computers and calculators. There are infinitely many numbers, like '1/3' and 'pi', that cannot be exactly represented internally (their decimal representations go on forever, and often cannot even be represented as a fraction, as in the example of 'pi'). As a result, any calculations that use these numbers are bound to be off.

5.2 Why does my HP say that 0 to the power of 0 equals 1?

Some claim that 0^0 is defined as 1, some claim it is indeterminate. The HP mathematicians decided to use the definition. Here is some more concrete info shamelessly lifted from the Sci.Math-FAQ:

Question:

What is 0^0 ?

Answer:

According to some Calculus textbooks, 0^0 is an "indeterminate form". When evaluating a limit of the form 0^0, then you need to know that limits of that form are called "indeterminate forms", and

that you need to use a special technique such as L'Hopital's rule to evaluate them. Otherwise, 0^0=1 seems to be the most useful choice for 0^0. This convention allows us to extend definitions in different areas of mathematics that otherwise would require treating 0 as a special case. Notice that 0^0 is a discontinuity of the function x^y.

Rotando & Korn show that if f and g are real functions that vanish at the origin and are analytic at 0 (infinitely differentiable is not sufficient), then f(x)^g(x) approaches 1 as x approaches 0 from the right.

From Concrete Mathematics p.162 (R. Graham, D. Knuth, O. Patashnik):

"Some textbooks leave the quantity 0^0 undefined, because the functions x^0 and 0^x have different limiting values when x decreases to 0. But this is a mistake. We must define:

x^0 = 1 for all x,

If the binomial theorem is to be valid when x=0, y=0, and/or x=-y. The theorem is too important to be arbitrarily restricted! By contrast, the function 0^x is quite unimportant."

Published by Addison-Wesley, 2nd printing Dec, 1988.

References:

H. E. Vaughan, The expression '0^0', Mathematics Teacher 63 (1970), pp.111-112.

Louis M. Rotando & Henry Korn, "The Indeterminate Form 0^0", Mathematics Magazine, Vol. 50, No. 1 (January 1977), pp. 41-42.

L.J. Paige, A note on indeterminate forms, American Mathematical Monthly, 61 (1954), 189-190; reprinted in the Mathematical Association of America's 1969 volume, Selected Papers on Calculus, pp. 210-211.

5.3 What is RPN?

Jan Lukasiewicz wrote a book on formal logic in 1951 that showed that mathematical expressions could be specified without parentheses by placing the operators before (Polish Notation) or after (Reverse Polish Notation) the operands. For example, the expression:

```
(4 + 5) * 6
```

Could be expressed in RPN as:

```
4 5 + 6 *
6 4 5 + *
```

Polish Notation is also known as prefix notation and Reverse Polish Notation (RPN) is also known as postfix notation.

5.4 Why did HP use RPN in their calculators?

In the years that followed Lukasiewicz's book, computer scientists realized that RPN or postfix notation was very efficient for computer math. As a postfix expression is scanned from left to right, operands are simply placed into a last-in, first-out (LIFO) stack and operators may be immediately applied to the operands at the bottom of the stack. By contrast, expressions with parentheses and precedence (infix notation) require

that operators be delayed until some later point. Thus, the compilers on on almost all modern computers converted statements to RPN for execution. (In fact, some computer manufacturers designed their computers around postfix notation.)

At the time that the HP-35 was introduced, other pocket calculators typically used a partial algebraic model. That meant they could evaluate trivial expressions like 4+5 but couldn't handle anything that involved parentheses or algebraic precedence. The technology of the time didn't allow for full algebraic compilers in pocket calculators.

RPN allowed HP to produce a pocket calculator that could evaluate arbitrary expressions using the available technology. For many, learning a new style of entry was a small price to pay to be able to evaluate arbitrary expressions on a calculator. Once the technology to produce algebraic compilers could fit into a pocket calculator, most RPN users had decided that RPN was more efficient and consistent for the user as well as for the calculator. Also, because subexpressions are evaluated as they are entered, entry errors are more obvious with RPN.

Another advantage to RPN is consistency between machines. Early algebraic models had differing limits of the complexity of the expressions they could evaluate. For example, TI catalogs from the late 70's listed how many levels of parentheses and pending operations each model could handle. Even today if you begin to use an algebraic calculator, you need to determine just "how algebraic" it really is. For example, some switch to RPN for unary operators (ie 5 SIN rather than SIN(5)) and some can't directly enter an expression like:

```
 4+5
-----
 6+7
```

5.5 Is there an alternative to calculating in RPN?

Many people just can't get used to the interaction with the stack to do calculations. A simple alternative is to place your equation in tic marks like '8+10*3' and to use the EVAL as an equals key. This way you can enter numbers just as you would write them on paper.

5.6 Why do I get unexpected values when I add temperatures?

Example: I add two temperatures like 34 deg.F and 11 deg.F, and I get 504.67 deg.F. Why don't I get 45 deg.F?

This occurs with the S/SX only. This behaviour changed in the G/GX. With the G/GX, adding 34 deg.F and 11 deg.F will give 45 deg.F using TINC and TDELTA. However, the G/GX will only add or subtract temperatures if the units are the same; it is now an error if they are not.

If the units attached to the number consists only of a temperature and nothing else, operations are done using an absolute temperature scale (relative to absolute zero). If the units consists of a temperature and something else, such as degrees/minute, the operation will not be done using an absolute temperature scale.

It also does not make any physical sense to add two temperatures. You can't add a cup of water at 20 degrees C to a cup of water at 30 degrees C and end up with two cups of water at 50 degrees C.

5.7 How can I add my own units? What's the 1_? unit for?

Note: Some later editions of the G/GX series manuals have some similar information on this subject.

Another feature is the generic unit, 1_?. It seems to be intended as a generic unit, to allow you to make new units not based upon any of the built-in ones. You could use it to create a conversion utility for currencies or bits and bytes. Before you create either, check the FTP sites to see what others have done in this regard.

- User units may be defined by creating a variable of any name (the new unit name), into which you store its definition in terms of other units, e.g. 2_mph 'stroll' STO; you can also use other user units in a definition, e.g. 3_stroll 'run' STO.
- You can then use the new unit name(s) in any custom or temporary menu, just the same as any other unit in the built-in Units Catalog. For example, { 1_stroll 1_run ... } TMENU

```
Here's an example. Convert this using ASC\-> and save it as 'sample'.

"69A20FF78E000000005046F6E6574750ADA209C2A2C2A2070000F368B01B2130
E20005046F6A756E650ADA203392010000000000000210C2A20F000046F6E6574
768B01B2130640005086F6D6562750ADA203392000000000000000510C2A20F00
0046F6E65747C2A20B0000D696E686B0168B01B2130B50003034354530 47A204
7A20C2A20F000046F6E65747ADA209C2A2C2A20F000046F6E6574768B01B2130
B213047A20C2A20F000046F6A756E6ADA209C2A2C2A20F000046F6A756E668B0
1B2130B213047A20C2A20F000086F6D65627ADA209C2A2C2A20F000086F6D656
2768B01B2130B2130B2130F0DA"
```

This is a sample directory with a custom menu. Enter the directory and press the CST key. Here you will see three units (truncated to fit the softkeys): donut, dozen, and homer. A dozen is defined as 12_donut, and a homer is a rate of 1.5_donut/min (not part of the SI system). These units may take SI prefixes described on page 10-5 of the User's Guide, so you may convert the value of 3_MW*h/kdonut into units of erg/mdozen using CONVERT, for example. The items in the CST menu have the usual abilities of unit menu objects as described on pages 10-2 and 10-3 of the User's Guide. These units can be used for math operations described on page 10-7 of the User's Guide, such as adding 2_donut to 1_dozen.

Example: At your power plant, you have four control room operators who have a combined consumption rate of 2.28 homers. How many donuts will they use in 5 minutes and 17 seconds?

- Enter the rate.

  ```
  2.28 [homer]
  ```

- Enter the time.

  ```
  Right-shift UNITS [TIME]
  5 [min]
  17 [s] +
  ```

- Multiply the time by the rate (*)
- Convert to dozens:

  ```
  CST Left-shift [dozen]
  ```

- The answer is a little over 1.5 dozen.

5.8 I want 'pi' to be a numeric value, not a symbol.

You are using "Numeric Constants Mode". To turn this mode on or off:

SX:

Type `-2 SF`. This turns the numeric constants mode on. To turn it off, type `-2 CF`.

GX:

Go to the MODES menu (press green/right-shift MODES). Hit the key for FLAG submenu. Toggle the second flag "Constant -> symb" to turn on and off.

5.9 What tricks are there for manipulating matrices?

SX:

You can build up a matrix by rows using \GS+. You can take a matrix apart by rows using \GS- These two commands (\GS is Sigma (Greek E)) add and subtract rows from the SigmaDAT Statistics matrix. \GS+ takes a vector row and adds it to SigmaDAT, and \GS- takes the bottom row away from SigmaDAT. \GS+ is accessible from the STAT menu with its own softkey. \GS- can be obtained by pressing left-shift /GS+, or by typing it in at the command line.

GX:

The GX has many matrix functions built in. See Appendix B-1.

5.10 Can I make my HP graph any faster?

Try lowering the plotting resolution by typing `#2 RES`. You can use larger values for faster speed at the expense of number of points plotted. In the GX Plotting Options Menu this setting is known as the "STEP" rate.

Alternatively, if you have a G series machine, you could try SpeedGraph, a fast assembly language plotter replacement. See `<http://www.hpcalc.org/>` for a copy.

5.11 How do I plot inequalities?

In the user's manual, inequalities are referred to as "truth plots". They are defined as expressions that return true (any nonzero real number) or false (0) results. In a truth plot, a pixel is turned on if the expression is true, it's unchanged if the expression is false.

For example to plot the simple inequality 'X>0' you could do:

```
'X>0' 'EQ' STO
TRUTH
-20 20 XRNG
-10 10 XRNG
ERASE DRAX DRAW
```

For more complicated examples see your manual, or on a G/GX you can type TEACH to look at some of the built-in examples. It may also be useful to plot the lines defining the outline true region. This is easily done by changing the plotting style to FUNCTION or CONIC and plotting again, taking care not to erase any previous plot.

Note that unless otherwise specified, every pixel in the display must be evaluated. As such, truth plots can be very slow. If you have a general idea on which values are going to return true, you can speed up plotting by specifying a smaller x and y plotting range. You can do this by setting the "LO" and "HI" input forms on a G/GX, which can be different than the actual display range.

If you are plotting the area between two equations, you speed up plotting dramatically by avoiding a TRUTH plot entirely. Store the two equations as the first two items in the current EQ list. Using NXEQ in the FCN menu will rotate the list as desired to plot both functions. Then the area between them can filled in very quickly using the SHADE key of the FCN menu (in the PICTURE environment).

Another hint for speeding up Truth plots is to set the resolution, say to 2 pixels or more, which also results in a different "shading pattern". The resolution can be set with the RES command.

5.12 How do I take the logarithm of a base other than 10 or e?

To take the log of a number other than base 10 or e, say log x base y, you use the formula:

```
  ln x
--------    where ln is natural log (log works too)
  ln y
```

A quick program to do it would be: << SWAP LN SWAP LN / >>

With the input of:

```
2: x
1: y
```

5.13 How do I solve quadratic equations?

The `QUAD` command is designed to solve expressions at degree up to two. One would expect the result of the simple expression `5=x/(1+x)` to be `x=-5/4`. But using the `QUAD` command, the HP gives the complex solution (0.5, 2.17944947177) i.e. the principal value (with flag 1 set). What is the reason for this?

The `QUAD` command is designed to solve quadratic equations. If the equation is not of the form:

```
a*x^2 + b*x + c = d*x^2 + d*x + e
```

The HP48 calculates a quadratic taylor series expansion which is then solved like a normal quadratic equation. The HP does the same when the equation is of a higher order. In the example above, the equation `x/(1+x)-5=0` has the expansion:

```
-x^2  + x - 5 = 0,
```

Which has two complex conjugate solutions. However, if you multiply the equation by `(1+x)/`, you get:

```
x = 5 * (1+x)
```

Which can be solved directly with QUAD. If the equation contains the variable only once, it´s the best way to use the ISOL command instead.

5.14 Why do I get a complex answers for roots of negative numbers?

This typically occurs for the (2n+1) roots of negative numbers, where n is a positive integer. For example, to calculate the fifth root of -3125 you can enter:

```
-3125^0.2
```

The result is the complex number (4.04508497187, 2.93892626146) or in DEG mode (4.99999999999, >36). The reason for this answer is that -3125^(1/5) has five different answers - arranged in a pentagon in the complex plane. In general, y^(1/x) has x different answers (y^x with x irrational has an infinite set of answers forming a circle on the complex plane).

Since 180/5 = 36 [degrees], this is the correct principal "5th root" of the negative argument in the complex plane; the HP48, unlike older calculators, does not turn "complex mode" on or off, so all "analytic" functions in the HP48 always accept complex arguments, and always return valid complex answers if suitable.

To calculate the desired solution, you have to use the right-shifted V-key with input like this:

```
2:        -3125
1:            5
```

Or you can use 'XROOT(5,-3125)' since XROOT is defined to both accept and return real values only.

5.15 Why doesn't '(-8)^(2/3)' give me the right answer (4)?

From: John Meyers

The reason is that the function ^ accepts and returns complex values whenever appropriate, and also always returns the "principal" answer in the complex plane, which may not be real if the argument is negative, even if there exist other real-valued possible results.

Here are some UserRPL programs which handle this sort of general "real power of real" question, even with fractional powers, including the sticky issue of getting the right sign.

Note that the "power" argument (level 1) may be either a fraction or a real number; we use \->Q in all cases to convert it to a fraction, which may be necessary to determine the correct sign of the answer.

```
\<< \->NUM DUP SIGN SWAP ABS \->Q
  IF DUP TYPE THEN OBJ\-> DROP2 ELSE 1 END
  4 ROLLD * ^ SWAP XROOT \>>
```

The version above is generally satisfactory for fractional powers which do not have a large numerator; otherwise an intermediate result might overflow, giving an invalid result.

The next version is overflow-resistant, but there is sometimes a slight roundoff error from using ^ ABS:

```
\<< \->NUM DUP2 ^ ABS SWAP ABS \->Q
  IF DUP TYPE THEN OBJ\-> DROP2 ELSE 1 END
  4 ROLL SIGN ROT ^ SWAP XROOT * \>>
```

Neither of the above programs ensures that its arguments are actually real; other arguments (e.g. complex) can sneak by, possibly producing meaningless answers. You can prevent this in either program by following \->NUM with OVER R\->B OVER R\->B DROP2

For example, with -8 and '2/3' as arguments, (or even -8 and .666666666667 as arguments), the answer is 4. To make the above into a user-defined function which can be used in algebraic expressions:

```
\<< \-> p r \<< r p \->NUM \->Q IF ... END \>> \>> 'XPOWR' STO
```

Then 'XPOWR(2/3,-8)' EVAL ==> 4 [using XPOWR() just like XROOT()]

Note: There are actually two different XROOT functions in the HP48; the algebraic version of XROOT expects its arguments in the opposite order from the stack-based version of XROOT (which inadvertently gave rise to a list-processing bug in early G-series ROM versions: K, L, M and P).

If you define XPOWR for yourself, as above, you will have to make sure that you use a consistent argument order for both algebraic and stack usage, unless you either make two separate functions or get into some fancy library-making, to duplicate something akin to the built-in "dual" XROOT functions.

5.16 How do I solve polynomials?

Another fast way to solve quadratic equations (or polynominals of higher degree) is by using the PROOT command. Place a vector with the coefficients of the polynominal on the stack.

For example:

```
x^3 - 9*x^2 + 26*x - 24
```

Would look like:

```
[1 -9 26 -24]
```

Then press left shift, SOLVE (key 7), POLY (key C), PROOT (key A). The result is [2 3 4], which are the roots of the equation.

5.17 How can I do statistics for grouped data?

Excerpt of postings from John Meyers

You must replicate the individual values or you can use a weighted analysis. For weighted analysis, Joe Horn's WEIGHT2 directory which can be found on Goodies Disk 8.

For weighted mean of numeric grades (or grade equivalents), with the number of credits for each course being the weight, you can use the program below.

First, fill in the statistics matrix, using any of the built-in statistics applications, entering each numeric grade equivalent in column 1, and the corresponding number of credits in column 2. Then press the key for the program below, which returns three results:

- Weighted mean of column 1, using weights from column 2
- Standard deviation (estimated from samples entered)
- Standard deviation (of population actually entered)

If you select an "X-Column" and/or a "Y-Column" different from the defaults (columns 1 and 2), then the results reflect the values in the "X-Column" using weights from the "Y-Column".

The program below is a slight improvement upon WEIGHT3, in that it no longer requires SigmaPAR to exist (the standard default columns are assumed if not). Credits go to Bob Wheeler for coming up with the algorithm, and to John Meyers for implementing the following program:

```
%%HP: T(3); @ \-> is right-arrow, \v/ is Square Root, \GS is Sigma
@ WMS - Weighted Mean and Standard Deviation, HP48G/GX
@ Stack diagram: ( SigmaDAT and SigmaPAR ) -->  Mean, SDev(S), SDev(P)
@ If SigmaPAR does not exist, then
@ Col 1 (XCOL) is value and Col 2 (YCOL) is weight, by default
\<< '\GSPAR' DUP VTYPE 0 < { DROP 1 2 } { RCL 1 2 SUB EVAL } IFTE
      0 0 DUP2 \-> x f w d m v
  \<< RCL\GS x COL- SWAP DROP ARRY\-> EVAL \->LIST
      RCL\GS f COL- SWAP DROP ARRY\-> EVAL \->LIST
      2 \<< DUP2 'w' STO+ m - * DUP 'd' STO
      w / 'm' STO+ m - d * 'v' STO+ \>> DOLIST
      m v w DUP2 1 - / \v/ ROT ROT / \v/
  \>> \>>^
```

If all you want is the weighted mean, and not the standard deviations, then you can dispense with all of the above, and just use a shortcut posted in GD7/MATH/WEIGHT.DOC by Joe Horn (basically, it is Sum(X*Y)/Sum(Y), where Y is the weight for X).

If you would rather replicate data points, then read on - however, if the "weights" associated with your data points are either very large or non-integers, then the following will be either very cumbersome or not possible, respectively. Press left-shift STAT DATA (on G/GX) to get to the menu containing the "Sigma+" "Sigma-" and "ClearSigma" commands.

Press "ClearSigma" to remove the old SigmaDAT. The first data must be entered as a vector: For example [2 4] and pressing "Sigma+". Now all other data pairs are entered without the brackets just as a pair separated by a space: e.g. 4 3 and pressing "Sigma+" and so on.

"Sigma-" in the HP48 does not accept data; rather, it simply removes the last-entered data point/row, popping it back onto the stack for you (the result is always one vector or scalar). If every point is removed, the statistics matrix is also deleted.

Now we can understand Joe Horn's program which accepts a data point/row, in whatever form the HP48 is prepared to accept it (single number, vector, or group of numbers), plus a frequency (integer only) on stack level 1, and then appends the data point/row the given number of times. Note that you must always supply a frequency, even if it is 1, and that any meaningless "frequency" less than 1 will be treated as 1.

```
\<< -55 CF 1 SWAP START \GS+ LASTARG NEXT DROP \>>
```

Obviously, \GS+ and \GS- represent the "Sigma+" and "Sigma-" commands.

Example: If the Statistics Matrix (SigmaDAT) already has two columns, and if we want to enter the new data pair [3 4] five times, we can simply type 3 4 5 and then press the key for the above program (don't forget to use a data vector for the very first data row).

For large frequencies, as noted, you will find this to be slow, not to mention the memory required for the multiple matrix rows; however, once you have literally entered every single data row, you can then do all the regression analysis and plotting you want, without the need for additional programming.

5.18 How do I generate a function table?

An easy and versatile way is to use the powerful list processing capabilities of the HP48. Generate a list with the x values, then perform the operations on that list. For example:

{ .1 2 2.5 5 123.4 } SQ will output { .01 4 6.25 25 15227.56 }

There are also several programs/libs to create a function table. Take a look at <http://www.hpcalc.org/> and search for "MathTools" by Jack Levy or "Table" by Diego Berge.

If you are low on memory or want a customizeable program you can use a tiny User-RPL program. Here are two suggestions with different forms of the output. All programs use the SEQ command, which is the key command here! (for further information read the manual). Input is for example:

```
5:    'SQ(x)'
4:         x
3:        11    (start)
2:        25    (end)
1:         1    (step)
```

The first mini program will output a matrix with the calculated values

```
%%HP: T(3)A(R)F(.);
\<< SEQ OBJ\-> COL\-> 1
COL\->
\>>
```

The second one will give x and f(x) in an array

```
%%HP: T(3)A(R)F(.);
\<< 4 DUPN 4 PICK 4
ROLLD SEQ OBJ\-> COL\->
6 ROLLD SEQ OBJ\->
COL\-> 2 COL\->
\>>
```

6 Questions about cards, data transfers, batteries, and hardware

6.1 My HP48 seems to making a high pitched noise.

From: Dave Arnett

The guts of the HP48 keyboard are assembled from several layers of mylar and similar material. Some of the layers have conductive ink to create switch contacts. Some are insulators, with appropriately located holes through which the switch contacts meet. Some are embossed with spherically domed areas, to provide the snap-spring feel when you press the plastic keys. The entire multi-layer sandwich is assembled onto plastic pegs on the backside of the HP48 top case, along with the metal chassis, and then the pegs are "heat stacked", positively capturing and holding the matrix together.

These parallel planes can act like a capacitive speaker. Most loudspeakers in home electronics are inductive: a sheet or cone moves because current passes through an attached coil positioned in a magnetic field. A capacitive speaker creates sound by moving a sheet or diaphragm due to electric charge forces, whether attracting or repelling. The HP48 keyboard has the necessary characteristics to be a very inefficient (therefore quiet) capacitive speaker.

When the HP48 "scans" the keyboard, charges are placed on the parallel sheets of mylar. They tend to move as a result of the charge, and create a faint squeal or whine. The loudness will depend on a number of factors, and can vary significantly from HP48 to HP48. Some folks report that, on their HP48, the sound is stronger on the back than on the front; others report the opposite.

6.2 Can I upgrade my S or G to more than 32K ram?

It is possible to do this by opening up the case and adding some memory chips. There are several containing instructions and schematics available at:

<http://www.hpcalc.org/docs/opening/>

There is a web page on how to open your HP48 from the back at <http://www.contrib.andrew.cmu.edu/drury/how2open.htm>

This is an illustrated guide on how to open your HP48 from the back. It has full instructions, as well as 15 pictures of the calculator in various stages of disassembly.

An alternate way to open your HP48 and upgrade it to 256K is described at the following site, along with lots of pictures:

<http://studwww.eurecom.fr/grundsch/>

Another good site is:

<http://www.cyberramp.net/elbert/hp48/>

WARNING: It is possible to ruin your HP in the process of performing any one of these upgrades. Do not even attempt it unless you are adept at working with small electronics. The FAQ maintainer and the authors of the instructions take no responsibility for anything you do to your HP!

6.3 Can I add a lithium battery backup?

From: Lee Studley *<RStud39515@aol.com>*

I recently did a modification to my 48G by replacing the 32K ram with a surface mount 128x8 RAM. Afterwards, I sandwiched 2nd and 3rd RAM chips by stacking them up and soldering. The 2nd CS was jumpered over to the expansion pin 21a. The decode for the 3rd part was easy to figure out. I've also added a lithium battery to insure memory retention. This thing now has 128K user memory, 128K in port 1, and 128K in port 2. Additional port 2 chips can easily be added.

The 3 volt lithium battery used was the generic coin type. I just made room for the holder and battery by removing the aluminum shield up to the plastic post near the piezo element. The positive terminal of the battery goes through a 470 ohm resistor into the anode of a schotky diode (only 0.2 volt forward drop). The cathode end of the diode feeds the positive side of the big 1000 uF capacitor near the added 74HC00. This is the memory retention cap when the batteries are removed.

This way, the lithium battery only gets used if the main batteries are removed or very low. Seems to work well so far. The ground connection broken by cutting the aluminum shield kills the reset and piezo contact points. To fix this, I stretched the original piezo contact spring and cut it in half. Then I soldered the cut piece to the ground "twisted" tab near the original spring. (readers should verify this ground with a meter.) This restored the lost functions.

6.4 How can I get/build a cable for my HP48?

Some places that sell the HP48 also sell the cable required for interfacing with a PC. Unfortunately, most don't.

For information on how to build your own cable, see Appendix D-1.

If you want to buy one, there are a few options. Places like ElekTek sell them for around $20-$25.

There are a few netters that make cables for sale also. Contact:

- Chris Edmunds <*chris@muffet.com*> Web page: `<http://muffet.com/hp/cables/ad.html>`
- Joel Kolstad <*kolstadj@peak.org*>
- Greg Lousteau <*greg@gregjeff.com*> `<http://www.gregjeff.com/greg/cable.html>`

If you live in a larger city, you can look in the Yellow Pages for a listing of HP authorized dealers who either have the cables in stock or can order them in about the same time a mail order place can, but for a lower price..

6.5 HP seems to offer two link kits with different prices.

HP offers two link kits, one which sells for about US$20, and the other which sells for about US$50. The only difference between the two is the higher priced kit comes with software and a manual. The cable, however, is the same in both kits.

As mentionned in the next question, free alternatives to this software are readily available. HP's software also includes some assorted miscellaneous programs and games, all of which are also available from HP via FTP or WWW.

6.6 Is there any communications software available?

In Windows 3.x you can use Terminal (in Accessories). In Windows 95 and OS/2 Warp, HyperTerminal can be used, or you can use any other "terminal emulator" package with some level of embedded Kermit file transfer capability.

However, the most complete functionality is available with full Kermit, which can be downloaded from <ftp://kermit.columbia.edu/> or HP. Columbia University is the original site for Kermit and has versions for nearly all operating systems, including all forms of Unix, Amiga, and Macintosh.

Of course if you have a G/GX machine you can use X-Modem instead of Kermit (X-Modem is generally faster). [Hyper]Terminal works fine for X-Modem as well. Alternative file transfer software is available from the FTP sites mentionned in the Appendix, or can be purchased (e.g. Donnelly's HP48 File Manager).

Also of interest are several freeware/shareware programs, including the HP48 Explorer for use with Windows 95. This can be obtained at <http://members.tripod.com/nberry/hp48g/index.html>.

6.7 Why doesn't the I/R port work farther than a few inches?

It turns out that it is the receiver that is "crippled". The transmit range is somewhere around several feet, and some people have actually written programs to make the HP48 emulate some remote controls (see Appendix E-1 for suggested programs).

There are currently two stories going around as to why. I will present both of them in the interest showing all sides of the story. You are free to decide which you believe:

1. The head of the HP48 design team, Dr. William Wickes himself, alluded during the Chicago Handheld Conference that the IR was intentionally crippled due to the concerns of certain individuals in academia that students could literally 'beam' information from one 48 to another during examinations. (from Rick Grevelle)
2. The reason the IR reception distance is so short is battery capacity. The Infrared Transmitter is semaphore: It is only turned on when it needs to be on, and the transmission protocol is defined to minimize the ON time. The receiver, by contrast, must be on continually when a transmission is in process, or when a transmission is expected. If you set your HP48 to Server mode, that receiver is drawing power. If you tweak the control bits to activate the receiver for any other reason, it is drawing power. The instantaneous peak power drawn by the transmitter is much higher than that of the receiver. But in the long run, the receiver will draw more energy from the batteries. Increasing the receiver sensitivity costs even more power.

6.8 Can my HP48 talk to my HP100LX via Infrared?

Yes, it can, although slowly. Set the HP100LX at 2400 bps, and put it in server mode (under the Kermit Commands menu). Then you can send stuff from your HP48 to your HP100LX.

6.9 Can I use my HP48 to communicate with IrDA devices?

From: David Johnson

I have had great success communicating via IR with my HP48 GX rev. R and a NEC Versa 6050MH notebook PC. However, I have found that speed is limited to 2400 bps. I have successfully used IR communication under Win95 with DOS Kermit 3.14, HyperTerminal, HP48 Explorer, Yellow Computing Transfile Win 48, and Kermit v0.85 (Win32).

I have had no success with Grab48 screen capture by HP or with HP PDL (only because I can't use a port higher than COM2). In my case DOS Kermit requires a command like:

```
set com3 \x03E8 \9     ; Syntax for nonstandard address or irq
```

Which is the port and IRQ specified in my BIOS setup. It doesn't seem to matter what IR speed I choose in the BIOS setup. However, in all cases (even native Win95 applications) I must disable the IR drivers in Win95 and talk straight to the port/IRQ specified in the BIOS. I can easily turn IR support on and off within a Win95 session with no ill effects via the control panel.

Note that the HP48 is not actually IrDA compatible, but communicates with IR devices in non-IrDA mode.

6.10 Can I print to a LaserJet printer with an HP48?

From: Derrik Pates

Yes you can, but you need the HP PCL library from `<http://www.hpcalc.org/utils/misc/pcl.zip>`. It allows you to communicate with any printer that supports PCL (all HP DeskJets and LaserJets, as well as some other laser printers and inkjet printers). You can hot-switch your resolution and turn PCL on/off with software commands.

Although newer HP printers such as the 5P and 5MP have an IR port, you cannot use it to print directly from your HP48.

6.11 I downloaded a program but all I get is "HPHP48-..."!

Some versions of Kermit distinguish between ASCII and binary files, and so you have to worry about the transfer modes at both ends of the connection (the HP48 SX end and the computer end). For example, when receiving ASCII files, Unix Kermit must translate CR/LF pairs into LF. If the HP48 SX is transmitting a binary file, but the Unix Kermit is expecting ASCII, any CR/LF pairs in the binary file will get translated to LF, corrupting the binary file. Unfortunately, you cannot "uncorrupt" the corrupted binary file by simply reversing the transfer and expecting Kermit to translate LF to CR/LF. This is because the binary file may contain occurrences of LF that were not originally part of a CR/LF sequence.

When a binary file gets corrupted, it will display on the stack as a string, starting with the characters "HPHP48-", and continuing on with a bunch of garbage. Thanks to two programs, these strings can be reconverted to binary files right on the HP48. These programs are: FIXIT, by Joe Horn and Mika Heiskanen, or OBJFIX, written by our friends at HP. These are included in Appendix A-2.

The major difference between FIXIT and OBJFIX is the way they operate. I have never personally had any problems with either. FIXIT takes the bad "HPHP48-" string from level 1 of the stack, and replaces it with the correct binary object. OBJFIX takes the name of the variable in which the "HPHP48-" string is stored from level 1 of the stack, and puts the correct binary object directly in the variable where the bad string was stored. It's up to you which you prefer. I would suggest reading the documentation of both before using either of them.

There is also a quick fix available from Dan Kirkland:

Here is a simple SYSEVAL program that will return a pointer to the object in those data transfer strings that start with "HPHP48-"

Checksum: # 8FEh

```
<< "12" SWAP + # 402Bh SYSEVAL # 62B9Ch SYSEVAL >>
```

That's all! (Simple huh!) Name? Whatever you want!! Works on all 48s (S/SX, G/GX)

6.12 What is the structure of HP48 ASCII headers?

From: Jorge Costa

ASCII file headers such as %HPHP: T(3)A(D)F(.); are created by the HP48 when an ASCII file is sent to the PC and read when a HP48 ASCII file is received from the PC. They serve to preserve various settings on your HP that affect ASCII transfers. They do not affect binary mode transfers.

Three parameters are saved: (T)ranslation Mechanism, (A)ngle Mode, and (F)raction Mark.

Translation Mechanism:

The possible ASCII translation schemes are as follows:

```
T   HP48 -> PC                  PC -> HP48
-----------------------------------------------------------
0   No translation              No translation

1   char 10 -> chars 10,13      chars 10,13 -> char 10

2   Action 1 & translate        Action 1 & translate
    chars 128 - 159             \000 - \159 to HP48 chars

3   Action 1 & translate        Action 1 and translate
    chars 128 - 255             \000 - \255 to HP48 chars
```

To choose your favorite set, just type the number followed by the command TRANSIO on the HP48 command line. Depending on which translation scheme you are using, the HP48 converts a different set of characters to PC ASCII format. I would advise using option 3, since this forces the calculator to translate all symbols to pure ASCII (look for the table in page 27 - 16 of the HP48G series user's guide). If you choose to use option 0 or 1, all characters from code 128 to 255 would be interpreted following the current code page of the PC setup - it will be unreadable!

Also, if you are receiving ASCII files in your machine and you get a bunch of black squares, you can be fairly certain you are using translation mode 0, which makes the HP receive and store all newline characters (instead of eliminating them).

Angle Mode:

- A(D) stands for degrees.
- A(R) stands for radians.
- A(G) stands for gradians.

Fraction Mark:

The HP48 can handle two types of fractions marks - the decimal dot or comma. This option is available from the MODES input form (48G/GX) or menu 64.01.

With regards to the ASCII header we have:

- F(.) means the format is dot.

- F(,) means the format is comma.

If you try manually editing an ASCII file on the PC without a header, then try to send it to the HP48 using Kermit, you can obtain the following results:

Number format	Text sent	Result (object stored)
.	3.4	3.4
.	3,4	3 4
,	3.4	3 4
,	3,4	3,4

As you would expect, sending files with ASCII headers to the HP48 with mismatching fraction mark settings, leads to an "Invalid Syntax" error during transfer.

Now take a look at rows 2 and 3 of the table above - the object received by the HP48 is stored as a program object without the << and >> composed by two real numbers - well, that normally can't be done in User RPL!

In fact the resulting object is a very simple System RPL program object (the internal language of the HP48). The conclusion is that when a file is sent without an ASCII header, the HP48 translates numbers with mismatching formats into System RPL secondaries (programs) of the form:

```
:: %IntegerPart %FractionalPart ;
```

6.13 Why do Kermit transfers seem to get slower?

From: Dan Kirkland

The reason the HP48 Kermit routines are so slow is because they are written in System RPL. Everytime the HP48 receives a packet (received as a string), it copies the previously received packets (string #1) and the new packet (string #2) to a new string (which becomes string #1 when the next packet is received). This is why the HP48 Kermit gets slower and slower on big transfers (string #1 just keeps getting bigger).

As to sliding windows... Sliding windows really don't help much with a clean, lag-free connection. And as most people connect their HP48 directly to the computer they are transfering to/from with a very short line, it is almost always very clean and lag-free. So, sliding windows have very little to offer HP48 transfers.

Yes, properly written Kermit routines should be at least as good as ZModem, even without sliding windows. And with a dirty, lagged connection, Kermit with sliding windows will blow ZModem away. But then, this just doesn't happen very often with HP48 connections.

Some claim that the HP48 can't do ZModem without tons of trickery because it doesn't have a large enough input buffer. While this is true, it shouldn't be too hard to do. And the same is true for Kermit if it is going to get ZModem type speed! Small packets are more reliable, but much bigger packets are needed for speed.

6.14 Why does XRECV not work sometimes? (GX)

Pre-Rev R. G Series 48's had a bug that would sometimes cause XRECV to fail if there was not twice the amount of room free for the incoming file. FXRECV, a fix for this bug, is available on the Horn 9 disk in the directory \HP as FXRECV. There is more info about this bug there as well. Note that FXRECV is not required for Rom R, and in fact will not run properly on Rom R.

6.15 Additional Tips on Transfers

To use the least memory when doing transfers, it is a good idea to use the built-in "Transfer..." application only to set parameters (Wire, Baud, etc...) but not to actually start a transfer with it; instead, press ENTER, then do either left-shift RECV (Kermit) or 0 FXRECV (Modified X-Modem). Avoiding the fancy "dialog box" application leaves more memory free at the time of actual transfer.

For best quick-start of a transfer, start the Kermit receive side before the transmit side. For X-Modem the opposite is true: start the transmit side before the receive side (with only a short interval between each side's start). Experiment a bit to find out what works best for you.

Here is a small program for Kermit that lets you receive directly to a port:

```
\<< -36 SF 1 \-> a \<< 'a' RECN a \>> SWAP STO \>>
```

For the argument use either just a number (if receiving a library) or a port-tagged name such as :0: name (for other objects, and in fact for libraries as well, but only the port number is used).

This program may be simpler to follow than the User's Guide directions for installing a library, since it encapsulates the several steps of storing in memory, recalling to stack, purging the variable, and finally storing to a port. Even if you forget to supply the portname argument, the received object will still appear on the stack (provided flag -55 is clear), and you can finish storing it afterwards.

6.16 What do the funny symbols in this document mean?

The symbols in this document such as \-> and \GS+ are the ASCII representation of the special HP48 graphical characters. See the section "Character Translations" in the manual for a complete table of symbols <–> ASCII representations.

SX:

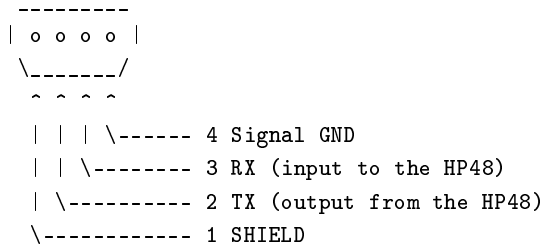
Chapter 33 (Volume II if 2 volumes)

GX:

Page 27-16 HP48 G Series User's Guide

6.17 What are the pinouts for the HP48 serial connector?

Looking at the pins of the HP48 (the following diagram is showing the pins on the HP48, as you look at the calculator):



```
Stated another way:

    HP48                    IBM 9 PIN               IBM 25 PIN
    (From the outside
     edge -> inward)
    SHIELD------------------  SHIELD----------------  SHIELD
    TX (Output)<-------------2 RX (Input)------------3 RX (Input)
    RX (Input)-------------->3 TX (Output)---------->2 TX (Output)
    SGND--------------------5 GND------------------7 GND
```

This information is also on page 27-7 of the G/GX manual.

6.18 Is there any information on interfacing to the HP48?

HP has made available an UNSUPPORTED document called the "HP48 I/O Technical Interfacing Guide", which contains information on the wired and serial I/O hardware. Another HP document available is the "HP48 I/O Software Interfacing Guide". Also, there are a number of guides contributed by users on making your own serial/IR hardware.

Look at <http://www.hpcalc.org/docs/programming/> as well as the ftp sites listed in the Appendix. If you find a good document, let me know so I can put it here!

6.19 How can I transfer programs/data from my HP-28S to my HP48?

You use a program called "INPRT", which is available on GD #9. Any previous version will crash the GX. The new version's checksum is #EDF3h, and the old, SX-only version's checksum is #2280h.

For each program that you want to transfer, you use the HP 28S to print it to the I/R output; INPRT, running on the HP48, reads the I/R output of the HP 28S and converts it into a program string.

6.20 Can I use rechargeable batteries with the HP48?

The Rayovac "Renewal" batteries have proven themselves to be reliable in the HP48, but there are some problems with Nickel Cadmium (Ni-Cad) batteries. You'll have to decide if using rechargeable batteries is worth putting up with the problems associated with them:

- You'll have to change batteries more often with rechargeable batteries. Alkaline batteries can deliver power for a longer period of time, compared to rechargeable batteries.
- Once the low-battery indicator comes on, you'll have to change the rechargeable batteries pretty quickly. With rechargeable batteries, you have only a few minutes before the calculator dies (it's been said that you have up to 15 minutes or so). With alkaline batteries, you have an amount of time that is supposedly measured in hours.
- Ni-Cad batteries lose their charge with time, even if you are not using them. Even if a Ni-Cad battery is just sitting unused on a shelf, it slowly loses its charge. Alkaline batteries keep their charge for a very long time.

6.21 How can I tell, from within a program, if the battery is low?

The following 48 program reports the status of the ALERT annunciator. The ALERT annunciator is activated by alarms or low battery.

This program is provided free of charge "as is" and has no warranty. No one is liable for any consequential damages. Preston

```
%%HP:T(3)A(D)F(.);
\<<
RCLF
8 STWS #FFh #0h +
#10Bh
#6595Ah SYSEVAL
#8h AND #0h >
SWAP STOF
\>>
```

6.22 I lost the information on my RAM Card when I changed the battery!

You forgot to turn ON the calculator before changing the battery. The HP only supplies power to the RAM card while the HP is turned on. While the calculator is turned off, no power is supplied by the calculator, and all power comes from the RAM card battery. If you then take out the battery from the RAM card while the calculator is turned OFF, the RAM card has no source of power, and you will lose all of the contents in the RAM card.

6.23 Why do I get an "Invalid card data" error when I merge a RAM card?

This message is usually seen when you plug a brand-new RAM card into an HP48. In this case, this message is normal and is harmless. It just means that there was no (valid) data on the card.

GX:

This message can appear when a RAM card contains ports you have never used. Press purple/left shift-LIBRARY. Hit NXT, then the PINIT softkey. This will initialize all available RAM ports. It does not affect data already stored on the card.

If you see this message under any other conditions, it could be one of the following:

1. You took the battery out of the RAM card while the card was out of the calculator, therefore erasing the memory.
2. You took the battery out of the RAM card while the card was in the calculator BUT the calculator was turned off. Note that the calculator supplies power to the RAM card ONLY when the calculator is turned ON. Again the memory on the card is gone.
3. The RAM card battery is dead.
4. There is a problem with either the calculator, the RAM card, or both.

If the battery is operational, then try to store something on the card. If you still get the error message, then backup all your info to a computer or another 48, and do a hardware reset.

6.24 I've heard about other manufacturer's RAM cards. Will they work?

While some cards may work, there is a chance that you may severely damage your HP48. You should only use cards specifically designed for the HP48. Naturally, HP sells such cards, and some third parties also make cards that will work in an HP48. Some cards work with both the S/SX and the G/GX. Some, however only work on the S/SX, or only on the G/GX. Make sure that the card is designed to work for your version before using it.

SX:

The S/SX can only naturally support up to 128K per port. HP sells 32K and 128K RAM cards. However, TDS, using software "bank switches" produces 256K and 512K RAM cards. These bank switching cards do not work in the G/GX series. CMT also makes 128K cards, which work in all HP48 calculators.

GX:

While you cannot use the TDS 256K and 512K cards, Sparcom has come out with 256K and 512K cards for the G/GX only. HP also sells 1MB RAM cards that work for G/GX only. The G/GX is able to accept a 4MB RAM card, but none are on the market at this time.

6.25 How does RAM card locking work?

From: Dave Arnett

The HP48SX had three layers of write protection:

- The Seiko-Epson memory cards contain write protection. Seiko-Epson is the OEM manufacturer for HP brand cards. They developed the format, connector and pin definitions for the memory card interface. We have a slightly modified version of their standard card where the low battery alarm limits are changed slightly.

  On an HP-branded RAM card, if you move the write-protect switch to the Protect position, two things happen. First, the card itself ignores write cycles. This is the first layer of defense for data integrity. The card also changes the voltage level of one of its output pins to tell the HP that it is write-protected.

- The HP CPU chip detects the write protect indicator line of the card. If it is reporting as unwriteable, the hardware locks out write functionality from the CPU chip. This is the second layer of defense. If this Card Detect line reports as writeable, then write transactions are permitted in hardware. If the Card Detect line floats, the hardware reports that no card is installed.

- The HP system firmware checks the status of the write protection bits before writing to a memory card. If the status bits indicate that the card is unwriteable (ROM or protected RAM), then the system code refuses to write to the card.

In developing the HP48GX, we changed card handling in some significant respects. I have recently posted information to the newsgroup on how the upper half of ROM and the card slot 2 are controlled through one pin of the Yorke CPU IC in a multiplexed fashion. There are some details about the Yorke chip related to how various Saturn Bus deviced configure and unconfigure which required me to make the following hardware assignment changes:

```
Hardware Resource           HP48SX          HP48GX

CE1- Card Controller 1      Card 1          Bank Select Controller
CE2- Card Controller 2      Card 2          Card 1
CE3- Card Controller 3      Unused          Card 2

CDT1- Card Detect 1         Card 1          Card 2
CDT2- Card Detect 2         Card 2          Card 1
```

Clearly, we had to assign Card Controllers and Card Detects in pairs, as much as possible. Otherwise the hardware lockout would make a mess of things.

A write protected RAM card in GX slot 1 has hardware lockout protection, since CDT2 can lock write actions to CE2.

A write protected RAM card in GX slot 2 does not have the same HP hardware protection. CDT1 is hardware coupled to CE1, which drives the Bank Select Controller. Some may have wondered in my recent posts why the protocol for setting up bank settings uses READ operations rather than WRITE operations. Now you know: A WRITE operation to the Bank Select Controller would be ineffective if a ROM or Write-Protected RAM card is in Slot 2!

The closest approximation to hardware write protection lies in the BEN (Bank Enable) line, which is bit 6 of the Bank setting. Unless BE is set high, all attempts to access slot 2 are forbidden. BEN is pin 10 of the 74HC174 chip. This line goes to pin 5 of the NAND chip to prevent Card Pin 21 from going active.

The protocol in the HP48GX RPL Opertaing System implementation is to always leave BEN low, unless slot 2 activity is needed. BEN should be set low again after card access is complete.

So an HP48GX has three levels of card write protection as well:

- The card should prohibit Write activity if the protection switch is set.
- Slot 1 has the same hardware lockout as in the SX. Slot 2 has a multi-step process necessary to enable access.
- The RPL system code checks the write-protect status before writing to a card. It understands that CDT1 and CE3 map to card 2.

6.26 What are the advantages and disadvantages of covered ports? (GX)

Covered ports are created when a card is placed in slot 2 of the GX, that is all ports >= 2. The advantage of this is very large RAM cards are possible: 256K, 512K, etc... up to 4 MB. Cards larger than 128K are split into multiple ports of 128K each controlled by a bank switching mechanism.

The disadvantages are that the memory cannot be merged with user memory (even if only a 128K card is used). Furthermore, some programs may be sensitive to the bank switching in covered ports, and may run slightly slower, improperly, or simply refuse to run at all. In particular, many of the freeware programs written in machine language are affected. Some contain special code to allow them to execute from both covered and standard ports.

6.27 Why does the HP48 display flicker slightly?

Display flicker is usually caused by fluorescent lights. The rapid pulsing of the fluorescent lights (60 Hz in the the U.S. and Canada), which is normally unnoticeable, interacts with the rapid pulsing/scanning of the HP48 LCD display (64 Hz refresh rate), which is also normally unnoticeable. The 60 Hz fluorescent lights alias with the display refresh (64 Hz) to produce a 4 Hz "flicker". It's normal and harmless.

Outside the U.S. and Canada, power is supplied at a 50 Hz rate, not 60 Hz. This means that the display "flickers" at a 14 Hz rate, which is still noticeable.

6.28 I broke the LCD screen - is there an easy way to get another?

Unfortunately, as far as I am aware, you cannot get a replacement, the cheapest alternative is to send the calc in for "standard service" at HP, which is slightly cheaper than a new calculator. Opening an HP48 is quite difficult anyway.

6.29 How can I protect my LCD?

There are many ways to protect the HP48's LCD. The solutions include:

- Inserting a piece of cardboard into the case.
- Cutting plexi-glass and putting it into the case (slightly better than cardboard). A piece of plastic cut out of a CD jewel case works too.
- Even better than plexiglass (because it's really unbreakable and will resist any twisting) is an electronic circuit board as used for computer mainboards etc... which should be available in any electronics shop. Another advantage to this method is the possible electrical shielding you get if there is a thin film of copper on one or both sides. I would recommend a board with the copper on one side only - on the side pointing to the outer side of the HP's soft case. After sawing it to the dimensions 79.5 mm x 179 mm x 1.25 mm and rounding the four corners (radius 4 mm), it will fit perfectly in the inner sleeve of the HP48's soft case and can also be used to hold a self-adhesive label with the owner's name. The small messages booklet will still fit, but it is better to store it on the outer side of the board so that the circuit board and the calculator will still fit together as tightly as possible (otherwise the booklet may break the LCD).
- Buying a hard case. A variety of types are available, including a hard leather case or a totally waterproof case.
- Keeping it in a video cassette or pencil case during transport.
- Using the cover from an old TI-8x calculator and fitting it into the grooves on the side of the HP48. Some people have had some success with this, but I find it barely fits.

6.30 Is there a rigid (protective) case for the HP48?

An inexpensive solution to this is to go to Sears and get a Black Plastic Tool Box #9 65283 in the hardware deptartment. It sells for only US $4.95. The case is big enough for the calculator in its case, plus extra batteries.

The case is a good fit for the calculator. If you want to go a step further, you can get some foam, glue it into the top and bottom halves of the case, and cut out a calculator shaped hole in the bottom. The result is lightweight, strong, secure and still leaves a little room for you to store extra batteries, a cable, and/or cards for your ports. Another solution is to use a cheque book as a case. Some cheque books just happen to fit the HP48, and provide a fair amount of protection.

The final alternative is to purchase a case. Eric Finley <*etfinley@ucdavis.edu*> sells protective hard cases for the HP48. They are stainless steel, very strong, and look nice. The cost is US $30. For more information, either e-mail Eric or check his web page at <http://www.ece.ucdavis.edu/etfinley/>.

6.31 Can I use my Amateur Radio with my HP48?

Yes, however you need to have a few things, the first being an amateur radio license. The next thing you need is a terminal node controller. If you are able to, get a terminal node controller that supports XON and XOFF software flow control. This is good to have because the HP48 only has a 256 character buffer. When the buffer is full, the HP48 sends the "buffer full" signal and a terminal node controller with software flow control can quit sending data until the HP48 can keep new data in the buffer. If you don't have a terminal node controller with software flow control, you can still use it, but you may lose some data.

The next thing you will want is some software to use with the terminal node controller. You could make your own software using user-rpl commands such as BUFLEN, SRECV, and XMIT. However, there is a plethora of software available. All you need is software then can send and receive data from the serial port. There are several programs at <http://www.hpcalc.org/utils/comms/terminal/> that you might like to try.

There is also a site that has a lot of hp48 programs relating to ham radio. Connect to <ftp://oak.oakland.edu/pub/hamradio/hp48/> Please inform the FAQ maintainer if you find additional software, so it can be added to this list.

7 Questions about programs

7.1 Where can I get programs and information for the HP48?

See Appendix E-1 for more info.

7.2 What are the Goodies Disks and where do I get them?

See Appendix E-5 for more info.

7.3 How do I get access to the HP Calculator BBS?

You can also access the BBS online at <ftp://ftp.hp.com/pub/calculators/>

You can also access the HP Calculator BBS via a modem (note that this is a long distance call for most people):

```
(208)-344-1691          2400 bps, 8N1
(541)-715-3277          9600 bps, 8N1
```

See Appendix E-4 for more BBS sites.

7.4 What are files that end with ".zip", ".Z", ".gz", or ".bz2"?

zip

Files that end with ".zip" are pkzip archives, which originated in the MSDOS world. One program that can extract the files in ".zip" archives is the InfoZip "unzip" program.

Binaries and source can be found at <http://www.cdrom.com/pub/infozip/>

gzip

Files that end with ".gz" have been compressed using the "gzip" compression command. To uncompress these files, you must use either the "gunzip" or "gzip -d" command. The "gunzip" program can also uncompress ".Z" files. Binaries should be on all Unix systems, but if not you can try <ftp://ftp.gnu.org/pub/gnu/>

More information should be there on binaries for other platforms.

compress

Files that end with ".Z" have been compressed using the Unix "compress" command. To uncompress these files, you can use the "uncompress" command on Unix. The "gunzip" program can also decompress these files.

bzip2

Another compression program gaining popularity is bzip2. bzip2 is a freely available, patent free, high-quality data compressor. It typically compresses files to within 10% to 15% of the best available techniques, whilst being around twice as fast at compression and six times faster at decompression.

Source and binaries for common platforms are available at <http://www.muraroa.demon.co.uk/>

7.5 What is a "ship" file?

In the past, programs for the HP48 posted on comp.sources.hp48 were posted in the "ship" format. This format made it easier for the moderator to distribute programs, but made it more difficult for users to extract and use these programs. Some people still use ship.

To extract programs distributed in the "ship" format, you need the following:

1. A copy of the "ship" program, compiled for your platform.
   - Sources for ship can be found at <ftp://ftp.cis.com/pub/hp48g/dos/unship.zip> which also includes the DOS executable.
   - Binaries for other platforms can be also be found at <ftp://wuarchive.wustl.edu/systems/hp/hp48/Posting/>
2. A copy of the latest InfoZip "unzip" program or an unarchiver that understands the pkunzip v2.04g format. Note that you cannot use older unzip programs, as old unzip programs do not understand the new compression formats. See the previous section for information on where to get source and binaries.

Once you have a copy of the "ship" and "unzip" programs, you use "ship" to convert the ship-encoded file into a .zip file, and you use the "unzip" program to extract the HP48 files from the ".zip" file.

7.6 What is the ASC format and how can I use it?

The functions ASC\-> and \->ASC were written by William Wickes to facilitate transferring HP48 binary objects in an ASCII format (useful when transferring objects via electronic mail or bulletin boards). Some HP48 objects, like libraries, cannot normally be converted into ASCII, unlike program objects, and these functions make it possible to do so.

The \->ASC function converts the object in level 1 into an ASCII string, which can then be uploaded to a computer for mailing.

The ASC\-> function converts the string object in level 1 back into an object. A checksum is used to ensure that the decoding is correct.

A copy of these programs are given in Appendix A-1, near the end of this file. Note however, that these functions aren't used much anymore. The standard in the newsgroup seems to first zip the programs together and post them uuencoded. But, you may still need these utilities for extracting older files you download.

7.7 What is the HYDE library, and how do I get rid of it?

The HYDE library is a library that changes the messages on your HP48. It works on all versions of the HP48 and is available on Goodies Disk 2. The program (HYDE.TXT) is in the DNICKEL directory needs to be converted with ASC\->. The instructions (HYDE.DOC) for the program are in the HORN2 directory. The library is 3679.5 bytes, small enough to keep on your HP48 so you can install it on a friend's when they're not looking.

Follow the previous instructions for installing the library, then go to the library menu and run HYDE.

However, getting rid of the HYDE library is a little trickier. First you must type JEKY on the command line. This will disable the HYDE library and restore the standard set of messages. Then you can follow the normal instructions for deleting a library.

7.8 What is the Minehunt game, and how do I use it?

The Minehunt game is a game built in to the G/GX. It was also included on the HP Equation-Library Card (for the HP48 SX).

GX:

It is located in the UTILS menu of the left-shift(purple) EQ LIB menu. (Left-Shift {EQ LIB], {UTILS}, {MINEHUNT}). See page 25-14 of the manual for a game explanation and picture.

SX:

It is located in the UTILS library.

You can save a game by pressing the STO button. This creates a variable with the name MHpar. If you are in the directory containing that variable, you will resume your old game when you run MINEHUNT.

To select the number of mines for your next game just store a number in a variable, Nmines (it's case sensitive), and you'll get that many mines. If you store a negative number, you'll get the absolute value of that many mines, AND all mines will be visible.

You can also move diagonally if you use the numbers as arrows: 1 is down+left, 2 is down, 3 is down+right, 4 is left, 6 is right, 7 is up+left, 8 is up, and 9 is up+right.

8 Questions about programming and advanced user functions

8.1 I've heard the names RPL, Saturn, STAR, GL etc... What are they?

ASAP

This is a simple Saturn assembler, written in the Perl language.

Chip8, Chip48, Schip, Schip8

This is a machine-code program that was inspired by the chip8 video game interpreter for the RCA CDP1802 microprocessor several years back. Chip8 allows you to write a simple graphics-based video game for the HP48 SX. Among the games written are "clones" of Breakout, PacMan, and Pong, to name a few.

Chip48, Schip and Schip8 are two different names for an enhanced version of chip8 specifically designed for the HP48 SX. People have written programs to assemble Schip assembly language into a form directly usable by Schip on an HP48. Schip interpreters are available for both G and S series.

GL

GL (Game Language) is a language for the programmer who wants almost the speed of assembly, but who doesn't want to spend a lot of time and effort to learn all the tricks of assembly programming on the HP48. GL has an assembly-like syntax, and there is an assembler for PC and one for the HP48 itself. Since GL is an interpreter, the GL library (3 KB in size) must be installed on the HP48 to run GL programs. The instruction set is powerful, with sprites and pixel based graphics, scrolling, sound, keyboard control, and more. There are 256 1-byte registers which can be used in pairs as 2-byte words. A GL program can be up to 64 KB in size. GL was inspired by SCHIP, but it is much better in all aspects. It is suitable for many applications, not only games, that require high speed. Of course it works on both the HP48 S/SX and the G/GX. GL is written by Erik Bryntse.

Java

Java is a library that displays an improved 5 level stack. It combines the best of SOL, HPSauce, EQStk, and others. Plus there are several new features and improvements! The intent of Java is to provide a more useful and faster stack environment for HP48 calculators.

The HP48 Java library has no connection at all to the Sun Microsystems product of the same name. Since the Java library is not a commercial product, we do not expect this to a problem.

Jazz

This is best System RPL and Machine Language development library that runs right on your HP48! Written by Mika Heiskanen. The Jazz library provides commands for assembling, disassembling and debugging both System RPL and machine language.

ML

Machine Language. This is usually used in reference to HP48 assembly language programming.

MLDL

Machine Language Development Library. This is a library that allows you to debug machine language programs with only an HP 48.

PDL

"Program Development Link". This is an DOS program, sold by Hewlett-Packard, that allows you to write and develop HP48 applications from your IBM PC clone. Programs are written on your PC, transferred to the HP48, and tested from your PC (and not the HP48).

RPL

RPL is the name of the language used to program the HP48 and HP-28 series calculators. RPL stands for "Reverse Polish Lisp". It's interesting to note that an HP Journal article incorrectly described RPL as "ROM-based Procedural Language".

From Bill Wickes:

RPL stands for Reverse Polish Lisp. In the early days of RPL development, we got tired of calling the unnamed system "the new system," and one of the development team came up with "RPL," both as a play on "RPN" which has been the loved/hated hallmark of HP calcs forever, and as an accurate indication of the derivation of the language from Forth and Lisp.

RPL was never particularly intended to be a public term; at the time of the HP Journal article (August 1987) on the HP 28C there was an attempt to create a less whimsical name–hence "ROM-based procedural language," which preserved the initials but had a more dignified sound. The development team never calls it anything but (the initials) RPL. You can choose either of the two full-word versions that you prefer. Or how about 'Rich People's Language?'

SASS

SASS is a simple Saturn assembler, written in C/yacc (BSD Unix & Bison). It uses Alonzo Gariepy's mnemonics.

Saturn

"Saturn" is the internal code name for the processor used in the HP48, HP-28 series, and many other calculators (just about all HP calculators since the HP 18/28 series).

STAR

This is the "Saturn Macro Assembler" (how "STAR" comes from this, I don't know), an assembler that uses mostly Alonzo Gariepy's mnemonics. It's written in C and runs on many different machines (PCs, Amigas, Unix, etc.). STAR is available via anonymous ftp from `<ftp://ftp.hp.com/pub/calculators/unix/star-1.04.4.tar.gz>`

System RPL

This is the name for the custom "operating system/language" used to program the CPU used in the HP48 calculator. System RPL is a superset of the normal user RPL commands; in addition to the normal, user-accessible RPL commands, system RPL includes many lower-level functions. System RPL is a language one step higher than assembly language.

HP has released a system development toolkit (for the IBM PC) containing a system RPL compiler, assembler, and loader, complete with a couple hundred pages of documentation. This toolkit, while copyrighted, is, for the most part, freely copyable. If you can find a BBS or archive site that has it, you can download all of the system RPL files and documentation.

USRLIB

USRLIB is a program that takes the contents of an HP48 directory and turns it into a library object. It currently runs under MSDOS.

Voyager

Voyager is an "interactive disassembler" for IBM PC clones that disassembles HP48 SX RPL and machine code. You download a copy of the RAM and ROM in your HP48 SX to your PC, and run Voyager. Using Voyager, you can then view the disassembled or unthreaded code. Unfortunately (or fortunately), Voyager uses the HP mnemonics, and not Alonzo Gariepy's mnemonics. Voyager is available from the HP BBS <ftp://ftp.hp.com/pub/calculators/>, and from various bulletin boards (the HP handhelds forum in CompuServe also has a copy). The latest version is "1.07".

8.2 Is there a C compiler for the HP48?

Yes. Alex Ramos has written a GNU C back end for the Saturn processor. Binaries are available for both Linux and Windows 95/NT, and source is available for compilation on other platforms.

Check <http://www.cris.com/ramos/hp48/hp48xgcc.htm> for the latest version.

8.3 Why do SysRPL programs run faster than UserRPL programs?

SysRPL is the built in language, custom designed for the Saturn processor used the HP48 family. UserRPL is a "subset" of SysRPL. The main reason for the speed difference is the fact that UserRPL commands have built in argument and error checking. In SysRPL, the programmer is responsible for all error checking to avoid memory clears.

Note that System RPL is a compiled language and that therefore you need a program like Jazz or GNU-Tools to create it. This means you can't edit or view SysRPL on the HP48 directly, unlike UserRPL. However both can be run the same way.

8.4 What is a good reference for learning SysRPL and ML?

Jim Donnelly's book "An Introduction to HP48 System RPL and Assembly Language Programming" provides 230 pages of step by step instructions for new programmers with plenty of examples. It also comes with a 3.5 inch disk containing the examples in source form and HP's development tools for DOS. It was previously available from EduCalc, I'm not sure if it is available elsewhere.

The ISBN number is: 1-879828-06-5

Also useful are RPLMAN.DOC and SASM.DOC available on Goodies Disk 4. See Appendix E-5 for more information. You might want to also have a look at the following URL: <http://titan.cs.uni-bonn.de/wolfrum/hpbooks.html>

Examples of ML programming can be found at:

- <http://www.geog.ubc.ca/amoy/>
- <http://www.ite.mh.se/danli97/>

8.5 Can I make my own libraries? Can I split others?

Yes, you can either use USRLIB in HP's development tools to create libraries in DOS, or you can make them right on your HP with hacking utilities like Rick Grevelle's D->LIB and L->DIR which can both create and split libraries from/to directories. These can be found in Mika Heiskanen's HACK library at <http://www.engr.uvic.ca/aschoorl/>

8.6 How do I know what library ID number to use for my program?

Note: While this document says HP48 SX, it is also valid for GX.

HP48 SX Library ID Numbers Paul Swadener, HP Developer Support. 03 Sep 1993

The HP48 will become confused if two or more libraries currently in the machine have the same ID numbers. To help third party developer efforts, HP maintains a list of HP48 Allocated Library ID Numbers.

Below are the "rules." The penalty suffered from not following these rules is that your library may not work if another library is in the machine with the same number.

1. Don't use any number in the range 000h through 300h. These are reserved for HP's use.
2. Don't use any number in the range 301h through 5A0h unless you have been assigned that number by HP. These numbers are reserved for developers who are in the business of distributing their software, that is, Independent Software Developers.

   To apply for the allocation of an ID# contact Paul Swadener, Hewlett-Packard Company, 1000 NE Circle Blvd., Corvallis, OR 97330, fax number 541-715-2192, internet address paul_swadener@hp6400.desk.hp.com. You must be a registered developer and supply the name/nature of your product, and current shipping address and phone/fax/internet add./etc.
3. Use a number in the range 5A1h through 600h for experimental work and software. HP does not track the use of these numbers.
4. Use a number in the range 601h through 6F6h for your personal applications, those which you will share, if at all, only to a known set of other users for personal uses. HP does not track the use of these numbers.
5. Don't use any number in the range 6F7 through 6FF, as these numbers were allocated before the formal allocation process in was instituted.
6. Don't use any number above 700h. These are also reserved for use by HP.

Hope this helps clear up how library numbers work.

8.7 What information is there on the internals of the HP48?

The most important documents are part of the "System RPL" development tools, a completely unsupported set of IBM PC tools created by HP Corvallis (the creators of the HP48). The tools, which run on DOS machines only, contain documentation on:

- System RPL (which includes information on HP48 SX display graphics, keyboard control, etc.).

- Saturn assembly language ("Saturn" is the name of the CPU in the HP48), including information on a Saturn assembler.
- A system RPL compiler.
- A Saturn object file linker.

However the documentation is also useful for some of the newer program development tools such as Jazz and GNU-Tools. (See the next question on where to obtain these).

8.8 Where can I get some programming development tools?

HP's unsupported System RPL development tools and documentation are available to customers free of charge to help them in HP48 application development, subject to certain legal terms, which are given with the tools (they're too long and detailed to give here).

You can get a copy in one of the following ways:

- Anonymous ftp to `<ftp://ftp.hp.com/pub/calculators/>`
- Get Goodies Disk 4. For info, see Appendix E-5.

Alternatively, you can use one of the packages put together by members of the HP community. In particular, Jazz is a library by Mika Heiskanen which allows you to program in System RPL and Machine Language directly on your HP48! It requires either an SX with a RAM card, or a GX.

Jazz is available at `<http://www.engr.uvic.ca/aschoorl/>`

There also is GNU-Tools by Mario Mikocevic (Mozgy) which is set of replacements for HP's development tools that not only runs on DOS, but also typical Unix systems such as Linux and Solaris. It is distributed in both source and binary form.

GNU-Tools is available at:

- `<ftp://gnjilux.cc.fer.hr/pub/hp48/gtools/>`
- `<http://www.zems.fer.hr/mozgy/jwz/hp48.html>`

8.9 I know UserRPL. How do I get started in SysRPL?

Here is a quick guide to learning System RPL. See the other questions for more information on where to obtain the items below.

- Learn UserRPL first, and become familiar with most commands.
- Get Goodies Disk 4 (HPTOOLS) or GNU-Tools if you have Linux etc...
- Get the new release of supported entries, available with GNU-Tools (not the entries in Goodies Disk 4).
- Read RPLMAN.DOC and/or get Jim's Donnelly book; it makes things easier.

- Get ENTRIES.SRT from the HP48 IRC channel.
- Get the tables from the SAD package by ftp to <ftp://ftp.hp.com/pub/calculators/>
- Get Jazz for your HP when you are ready.
- You can learn a lot by studying the ROM and other people's programs with Jazz!

8.10 Are there any viruses for the HP48?

There are several, and most are French. In the United States, a virus called the "Michigan virus" erupted a couple of years ago. There is a program on Goodies Disk 8 that checks for the Michigan Virus. Currently, viruses do not pose an appreciable threat on HP48's.

8.11 How do I store fields of variable length effectively?

See Appendix A-5 near the end of this file.

8.12 What is "Vectored Enter", and how do I use it?

From: Bill Wickes

The HP48 manuals do not document a very powerful feature that we call "Vectored ENTER," that allows you in effect to redefine or bypass the command line parser and to have a shot at the stack etc. after the command line has been executed.

Keys that execute an automatic ENTER perform a two-step process:

1. The command line is parsed and evaluated.
2. The key definition is executed.

When flags -62 and -63 are both set, the system extends this process as follows:

1. The current path is searched for a global variable named *aENTER (where "*a" is the Greek alpha character–character 140). If present, the command line is entered as a string object and *aENTER is executed. If absent, the command line is parsed and evaluated normally.
2. The key definition is executed.
3. The current path is searched for a global variable named *bENTER ("*b" is Greek beta–character 223). If present, then a string representing the key definition is put on the stack, and *bENTER is executed. The string is the key definition object's name if it is a command, XLIB name, global or local name, or an empty string for other object types; its primary purpose is to implement things like the TRACE mode on other calcs, where you can print a running record of what you do.

A simple example of the use of *aENTER is to create a more convenient binary calculator, where *aENTER slaps a "#" on the front of the command line so you don't have to bother when entering numbers.

8.13 What is "WSLOG"?

It is an until recently undocumented feature which stands for "Warm Start Log" (it is listed in the command appendix of newer G manuals, as well as being fully explained in the AUR).

Type in WSLOG in caps, and calc will list the time and cause of the last four warm starts. This feature helps HP technical support to fix your HP48 in case it crashes.

```
0 - log cleared by <ON> <SPC> then <ON>
1 - low battery condition, auto deep sleep mode invoked to save battery
2 - hardware failed during IR (time out)
3 - run through address 0
4 - system time corrupt
5 - deep sleep mode wake up (alarm?)
6 - unused
7 - CMOS test word in RAM was corrupted
8 - abnormality was detected about device config
9 - corrupt alarm list
A - problem with RAM move
B - card module pulled
C - hardware reset
D - system RPL error handler not found in run stream
E - corrupt config table
F - system RAM card pulled
```

8.14 What are SYSEVALs?

SYSEVALs are addresses that point directly to a location in the HP48's ROM. Many SYSEVAL hex strings are System-RPL commands. Because there is no argument checking, it very easy to clear your HP48's memory.

In the file SYMBOLS.GX, HP designates three types of entries. The first type is "supported." This means that a particular entry will work on ALL versions of the HP48 and HP supports the use of this entry.

The second type of entry is "unsupported." This means that HP does not support the use of this entry and it will not be the same on all versions of the HP48.

The third type is "unsupported-static." This is an entry to HP does not support the use of, but it is the same for all versions.

8.15 What are some useful SYSEVALs?

For a complete list of supported entry points, see <http://www.engr.uvic.ca/aschoorl/>

In System-RPL, different HP object types are referred to by symbols. These symbols are used to construct stack diagrams, which tell the user how to use System-RPL commands. Here are some of the symbols:

```
Symbol  Type                    Example
------  --------------------    -----------------
$       string                  "ABCDEFG"
id      Global Name             'A'
```

```
arry    Array                    [[ 1 2 ] [ 3 4 ]]
grob    HP48 Graphics Object     Graphic 131x64
#       Binary Number            <1h>
symb    Symbolic Object          'A^2+B^2'
```

To interpret the stack diagram:

Everything on the left of the -> arrow is the input. That needs to be on the stack before executing the SYSEVAL. The symbols show you which kinds of data to put on the stack. Once you execute the syseval (see next paragraph), the HP will return the data back that is indicated on the right side of the arrow.

To use the following entries, do the following:

1. Make sure (triple check) that the appropriate arguments are on the stack.
2. Push Right-shift /, you should get a # on the command line.
3. Enter the hex string exactly as shown in the hex string column.
4. Put an small case h at the end of the hex string.
5. Press enter.
6. Type SYSEVAL.

Here are just a few examples:

```
Hex    Sys-RPL Name    Category            Stack Diagram
---    ------------    --------            -------------

15777  NULLID          Supported           ( -> '' )
30794  VERSTRING       Supported           ( -> "HPHP48-?" )
3A328  MakeStdLabel    Supported           ( $ -> grob )
3A3EC  MakeDirLabel    Supported           ( $ -> grob )
3A38A  MakeBoxLabel    Supported           ( $ -> grob )
3A44E  MakeInvLabel    Supported           ( $ -> grob )
3A1FC  DispMenu.1      Supported           ( -> )
05F42  GARBAGE         Supported           ( -> )
41F65  WaitForKey      Supported           ( -> #KeyCode #Plane )
353AB  SYMB>IDS        Unsupported-Static  ( symb -> list )
40D25  LockAlpha       Supported           ( -> )
40D39  UnlockAlpha     Supported           ( -> )
3AA0A  1A/LockA        Supported           ( -> )
44C31  DoNewMatrix     Supported           ( -> arry )
44FE7  DoOldMatrix     Supported           ( arry -> arry' T/F )
1314D  TOADISP         Supported           ( -> )
13135  TOGDISP         Supported           ( -> )
39531  ClrDA1IsStat    Supported           ( -> )
130AC  RECLAIMDISP     Supported           ( -> )
4E347  TURNMENUON      Supported           ( -> )
05B15  $>ID            Supported           ( $ -> id )
05BE9  ID>$            Supported           ( id -> $ )
3A1E8  DispMenu        Supported           ( -> )
39BAD  DispStack       Unsupported-Static  ( -> )
```

8.16 What are LIBEVALs?

From: Joe Horn (Copied from "libeval.doc" on GD 9)

The G/GX has a new command: LIBEVAL. It's sort of like SYSEVAL, but instead of taking a memory address (which could be ambiguous due to the G/GX's massive need for bank switching), it takes XLIB numbers. It's a "back door" for us hackers. Unlike the ELSYSEVAL command in the HP Solve Equations Library Card, which only worked for the libraries in that card, LIBEVAL works for any library, including the hidden ones in the operating system.

> CAUTION: LIBEVAL, like SYSEVAL, is both powerful and dangerous. If used with incorrect input(s) and/or in the wrong environment, it can nuke your memory.

LIBEVAL takes a single user binary integer of the form #LLLFFFh, where LLL is the library ID (hex), and FFF is the three-digit function number (hex, with leading zeros if necessary).

For example, the R->B command is XLIB 2 9. So you can perform a R->B by executing #2009h LIBEVAL. Try it: 10 #2009h LIBEVAL -> #Ah.

Using it for named commands is unnecessary, but the G/GX is highly XLIB oriented (again, due to the need for bank switching), and there are many useful and interesting features accessible only through LIBEVAL.

A prime example is the G/GX's inability to programmatically launch the new "dialog boxes". Suppose a program needs to print things; it would be awfully nice to be able to throw the I/O PRINT dialog box onto the screen. The User's Guide is no help here, and in fact it would seem impossible to do. But #B4197 LIBEVAL is all it takes! Try it. Cool, huh?

8.17 What are some useful LIBEVALs? (GX)

See Appendix B-3 for a few GX LIBEVALs.

8.18 What is the format of a GROB object?

Note: I received two very good answers to this question, and each cater to a different kind of user, so I will include them both.

The easy way:

First, add pixels to the right-hand edge to make the horizontal dimension a multiple of 8, then separate the object horizontally into groups of four pixels. Suppose, for example, you want to make this into a 22x8 GROB:

```
---------****---------
--------**--**--------
-------**----**-------
********------********
********------********
-------**----**-------
--------**--**--------
---------****---------
```

You would break it up like this:

```
---- ---- -*** *--- ---- ----
---- ---- **-- **-- ---- ----
---- ---* *--- -**- ---- ----
**** **** ---- --** **** **--
**** **** ---- --** **** **--
---- ---* *--- -**- ---- ----
---- ---- **-- **-- ---- ----
---- ---- -*** *--- ---- ----
```

Notice, I've added two columns of pixels to make it 24x8. Now, each group of four can be represented by a hexadecimal digit.

```
---- 0          --*- 4          ---* 8          --** C
*--- 1          *-*- 5          *--* 9          *-** D
-*-- 2          -**- 6          -*-* A          -*** E
**-- 3          ***- 7          **-* B          **** F
```

Now, replace each group of four pixels with its corresponding hexadecimal digit: 00E100 003300 081600 FF0CF3 FF0CF3 081600 003300 00E100.

So, your final grob is:

```
GROB 22 8 00E100003300081600FF0CF3FF0CF308160000330000E100
```

The technical way:

1. Every HP object format is described in RPLMAN.DOC, see this file for more info.
2. All HP objects are (of course) nibble based. When viewing them on the PC, which is byte based, the low order nib is always read and written first.
3. HP objects begin with a 5 nib prologue which identifies the object type (and other things). This is the prologue for a grob object..

   ```
   02B1E pr HP:DOGROB
   02B1E @  Graphic prologue (/U GROB)
   ```

4. Every 5 nib sub field of an object, such as prologue, length, etc., is read in reverse order (low nib first). The grob prologue would appear E1B20, or 1E 2B x0 in a hex dump on a PC. The least significant nibble of the next field begins right at the x in x0.
5. It is customary to always work with HP objects on the PC with the HPHP48-E header (substitute your favourite rom version letter) preceeding the prologue. This gives no hassles when downloading via Kermit or X-Modem.

PC hex dump of a grob object.. (remember each 5 nib field is reverse order)

1. "HPHP48-E" 8 byte ascii string with msb off
2. 5 nib prologue, 02B1E (hex)
3. 5 nib length field, includes length of body AND length and height fields! does not include 5 nib prologue. So, the total #nibs the grob object takes in HP memory is this number + 5. Important!
4. 5 nib height field (yes, it is height then width)

5. 5 nib width field
6. body (described below)

The grob object looking at it using command line EDIT on the HP..

```
GROB <width (dec)> <height (dec)> < body >
```

The body is exactly the same looking at it on a PC hex dump or on the command line EDIT. Just remember to read low nibble first for hex dump.

Because of hardware restrictions, the number of nibs required to represent one horizontal line must be even (byte aligned). So, use this method to calculate the number of nibs in one line.. (pascal example)

```
nibs := width div 4;
if nibs mod 4 <> 0 then nibs := nibs + 1;
if odd(nibs) then nibs := nibs + 1;
```

Then to get the #nibs in the entire body, multiply it by the height. Remember, the length field must be this total body length + 15 !

For example, a 131 wide grob requires 34 nibs for one horizontal line, where 5 of the bits are unused. The first nib in the second line begins at nib 35 for this grob, etc... If the grob is 64 lines, then the body is 2176 nibs. Add 5 for the prologue, 5 for the length field, 5 for height, and 5 for width. Total object size (on the HP) is 2196 nibs, or 1098 bytes. The length field should contain 2196 minus 5 for prologue = 2191 (0088F hex)

Note that for each nib, the right-most bit is the left-most pixel on the screen. This is nearly always the case for graphic memory.

8.19 What is the AUR and what information does it contain?

From: Jarno Peschier <*jpeschie@cs.ruu.nl*>

In this text I will try to answer a number of frequently asked questions about the HP48 G Series Advanced User's Reference Manual. In the remainder of this text I will use the common abbreviation AUR when referring to this book.

A good link for AUR information is at: `<http://www.contrib.andrew.cmu.edu/drury/hp48gaur/>`

The questions I will try to answer are:

- What is the AUR?
- What information does the AUR contain?
- Do I need the AUR?
- Why didn't HP supply the AUR with every HP48?
- Where can I get the AUR?
- Does the AUR contain any SYSEVALs and/or LIBEVALs?
- Does the AUR say anything about SysRPL and/or ML?

What is the AUR?

The AUR is just what it's name tells you it is: it's a reference manual for the HP48 G Series. It has the same outside appearance as the HP48 G Series User Guide (the manual you get with your HP48 G or HP48 GX when you buy it), except that it's much thicker (just under 4 cm). The AUR contains about 650 pages of useful information about your calculator.

What information does the AUR contain?

The AUR contains information about UserRPL programming, quite a lot of programming examples, a complete command reference for all the (UserRPL) commands the G Series knows and reference lists about all equations, error and status messages, units, etc... Of course it has an extensive index at the back of the book.

Chapter 1: Programming

Contains: Understanding Programming, Entering and Executing Programs, Viewing and Editing Programs, Creating Programs on a Computer, Using Local Variables, Using Tests and Conditional Structures, Using Loop Structures, Using Flags, Using Subroutines, Single-Stepping through a Program, Trapping Errors, Input, Stopping a Program for Keystroke Input, Output, Using Menus with Programs, Turning Off the HP48 from a Program.

Chapter 2: Programming Examples

Contains: Fibonacci Numbers, Displaying a Binary Integer, Median of Statistics Data, Expanding and Collecting Completely, Minimum and Maximum Array Elements, Applying a Program to an Array, Converting Between Number Bases, Verifying Program Arguments, Converting Procedures from Algebraic to RPN, Bessel Functions, Animation of Successive Taylor's Polynomials, Programmatic Use of Statistics and Plotting, Trace Mode, Inverse-Function Solver, Animating a Graphical Image.

Chapter 3: Command Reference

This chapter contains one entry for every command (except for RULES) from ABS to ZVOL and + to ->. Each entry contains:

- A description of what the command does (with stack diagrams).
- Ways in which you can access it from the keyboard (with submenu you need, or if you can only type it in in Alpha-mode).
- Which flags the command is affected by.
- Remarks, and a list of related commands.

Often a command has one or more example programs too. This chapter is the bulk of the whole book; it's 424 pages thick.

Chapter 4: Equation Reference

This chapter contains one entry for every section in the built in Equation Library (EQLIB). Each entry contains: a table of used variables and then for every set of equations an entry with some more info about the equation set, the picture that goes with it (if any), all the equations and one or more sets of example numbers and solutions. The end of the chapter has a list of references from which all the equations were taken.

Appendix A: Error and Status Messages

This chapter contains two tables with all possible messages. The first has them sorted alphabetically and shows the message, the meaning and the ERRN number in hex. The second has them

sorted on ERRN number (and thus on category) and only contains the number in hex and the message itself.

Appendix B: Table of Units

This chapter contains a table with all units the HP48 G Series knows. It contains the unit (both the name you use on the calculator as well as the full name) and the value it represents in SI units.

Appendix C: System Flags

This chapter contains a table with all the system flags of the HP48 G Series. The table contains the number followed by a description of it's use and the results you get when the flag is set or clear.

Appendix D: Reserved Variables

This chapter contains a table with all the reserved variables of the HP48 G Series (ALRMDAT, CST, "der"-names, EQ, EXPR, IOPAR, MHpar, Mpar, n1/n2/..., Nmines, PPAR, PRTPAR, s1/s2/..., VPAR, ZPAR, SigmaDAT and SigmaPAR) with complete information about what they are used for and about all possible parameters that can be put into them.

Appendix E: New Commands

This chapter lists all the commands that are new to the HP48 G Series, with a brief description of what the commands do. A list like this can be found elsewhere in the HP48 FAQ list as well.

Appendix F: Technical Reference

Contains: Object sizes, Mathematical simplification rules used by the HP48, Symbolic differentiation patterns used by the HP48, the EquationWriter's expansion rules, References used as sources for constants and equations in the HP48 (other than those in the Equation Library).

Appendix G: Parallel Processing with Lists

This chapter contains info on parallel processing that makes UserRPL look even more like a real functional programming language by letting (almost) all the commands function on lists of parameters as well (i.e. adding map capabilities to all internal commands).

Do I need the AUR?

Yes, you do. In my humble opinion the AUR is just "The Manual, Part II" and every owner of a HP48 G or HP48 GX should have it, especially if you are (going to be) programming in UserRPL for any reason. A lot of the frequently asked questions that pop up on comp.sys.hp48 are simply questions that are meant to be answered by the AUR.

Why didn't HP supply the AUR with every HP48?

Since you're reading this FAQ, you're probably gifted: you are not an average HP48 user. The set of "average users" probably consists mostly of people that simply never program their HP48 in any way ("Wow, can it do that too?!"). In this case, they will never take one look at the AUR ever again, and since this is quite a thick book it would be a waste of money, trees, the environment, transportation costs, etc... to supply the AUR in every box. This is probably why HP made it an extra accessory for those people that "really need" it. I think...

Where can I get the AUR?

You should be able to buy (or order) it from the same shop where you purchased your calculator. The HP part number is 00048-90136. The book does not seem to have an ISBN; it's a HP part, not a book.

Does the AUR contain any SYSEVALs or LIBEVALs?

Yes, it contains exactly one. The example for the SYSEVAL command tells you that #30794h SYSEVAL gives you the version of your HP48. My HP48 GX rev. P gives "HPHP48-P".

Does the AUR say anything about SysRPL or ML?

No, it doesn't.

8.20 What is the syntax for INFORM, CHOOSE, and MSGBOX? (GX)

Based on a posting on HPCVBBS by Cary McCallister. Revised by Joe Horn. Re-formatted by Andre Schoorl.

INFORM - Create a Custom Input Form

INPUT:

```
5: "title"
4: { fields }
3: <format>
2: { reset_values }
1: { initial_values }
```

OUTPUT:

```
Either: 2: { new_values }
        1: 1

or:     1: 0
```

PURPOSE:

INFORM creates full-screen custom input forms, similar to those of the built-in applications.

ARGUMENTS:

title

Is displayed in font 1 (small) at the top centre of the screen. It may contain any number of characters, including 0, but, at most, 31 characters will be displayed. Strings longer than 31 character will be truncated to 30 characters and displayed with a trailing ellipsis. Line feed characters are displayed as boxes.

{ **fields** }

Is a list of the form { field1 field2 ... fieldn } where each field is any one of these forms:

```
1.    "label"
2.    { "label" "help" }
3.    { "label" "help" type1 type2 ... typen }
4.    { }
```

The "label" is a string of arbitrary text from 0 to 26 characters long (longer strings will be truncated to 25 characters with a trailing ellipsis). The "help" is displayed immediately above the menu area as a helpful prompt for the user. The object type specifications list the one or more object types allowed in the field, in the form the TYPE command returns. If

unspecified, the default help text is blank and the field will accept all object types. If a field specification is an empty list, no field is produced; instead, the field to the immediate left (if any) is expanded to fill the omitted field's space. This allows further customization of form layout; for example, to have two fields in one row and three fields in the next.

<format>

May be any one of the following objects:

1. `{ }`
2. `columns`
3. `{ columns }`
4. `{ columns widths }`

Where <columns> is the number of columns that the display will be divided into e.g., if <columns> is 2 and there are two fields, they will be displayed side by side rather than one above the other. <widths> is the tab width between the left edge of each title and its field; this makes vertical alignment of fields possible. Titles longer than one tab width will cause the tab to be repeated automatically. If unspecified, the default column count is one and the default tab stop width is three.

{ **reset_values** }

Is a list of values one per input field which replace the values of all fields when the {RESET} key is pressed and "Reset all" is selected. This list must either be empty or have exactly one item per field corresponding to the { fields } list above. If empty, then all fields are deemed unspecified (i.e., blank). Each reset value must match the type require by the form field. Selected fields may be marked as unspecified by using NOVAL as the reset value.

{ **initial_values** }

is a list of values one per input field which specify the start-up values of all fields when the INFORM command is invoked. This list must either be empty or have exactly one item per field corresponding to the { fields } list above. If empty, then all fields are deemed unspecified (i.e., blank). Each initial value must match the type require by the form field. Selected fields may be marked as unspecified by using NOVAL as the initial value.

RESULTS:

INFORM returns the new field values { new_values } as a list to level 2 and the real value 1 to level 1 if the user completes the input form by pressing {OK} or [ENTER]. The list will contain one item per field. Each item will either be the last input value for the corresponding field, or NOVAL if the field was unspecified.

INFORM returns the real value 0 to level 1 if the user terminates the input form by pressing {CANCL} or [CANCEL].

REMARKS:

NOVAL is basically a command (an XLIB name) that does nothing when evaluated. It can, however, be compared via ==, SAME and POS.

CHOOSE - Create a User-Defined Choose-Box

INPUT:

```
3: "title"
2: { items }
1: <initial_item_number>
```

OUTPUT:

```
Either: 2: { chosen_item }
        1: 1

or:     1: 0
```

PURPOSE:

CHOOSE creates a user-defined "choose-box": a displayed box containing items from which the user can choose one.

CHOOSE displays a standard choose box (normal, not maximized; single-pick, not multi-pick) with an optional title line.

ARGUMENTS:

title

If any, is displayed in font 1 (small) in the top centre of the box. It may contain any number of characters, including 0, but, at most, 22 characters will be displayed. Strings longer than 22 character will be truncated to 21 characters and displayed with a trailing ellipsis. Line feed characters are displayed as boxes. If there are more than four items such that the scroll arrows are displayed, the maximum number of title characters is reduced to 21. The title text is displayed in the title area in character font 1 (the menu font). If the title string is empty, no title area will be created in the choose box, and the whole box will be devoted to the items.

{ **items** }

Is a list of arbitrary objects. Each item occupies one line in the choose box and is displayed in its command line form. Only the first 14 characters of the displayed object will be shown (13 if the scroll arrows are present). If an item contains more than 14 (13) characters, it will be truncated to 13 (12) and the final character will be replaced by an ellipsis. If every item is a list of exactly two objects, the first object will be displayed and the second returned if the item is selected. If the number of items exceeds 4 with a title and 5 without, scroll arrows will be displayed, and moving the highlight past the top or bottom of the box will scroll through the undisplayed items.

<initial_item_number>

Specifies the initial field to be highlighted when the display first comes up (ordinarily 1). A value of 0 indicates that no field is to be highlighted and that the {CANCL} key is to be omitted from the menu, making the choose box act as an informational display without user selection.

RESULTS:

CHOOSE returns the chosen item (or the second object in a two-element item list) and the real number 1 if the user completes the choose box by pressing {OK} or [ENTER]. CHOOSE returns the real number 0 if the user terminates the choose box by pressing {CANCL} or [CANCEL].

MSGBOX - Create a Simple Message Box

INPUT:

```
1: "Message_string"
```

OUTPUT:

```
None
```

PURPOSE:

MSGBOX creates a simple pop-up message box over the current display, with a 3D shadow to make it easy to recognize, and containing any arbitrary text.

ARGUMENTS:

Message_string

Is any string of characters. The text is displayed in font 2 (medium size), left justified, with a maximum line length of 15 characters. Longer lines are broken into multiple lines automatically, at word breaks when possible, up to a maximum of 5 lines. Line feeds may be included to force line breaks. The minimum number of lines is 2, so single-line messages will have a blank line at the bottom of the box.

RESULTS:

None. MSGBOX is exited when the user presses CANCEL, ENTER, or OK, but nothing is returned to the stack.

For examples of INFORM, CHOOSE, and MSGBOX see Appendix B-2.

8.21 How do I put checkfields in my INFORM menus?

From: Matt Willis

You can't normally do this directly in User RPL. You can do it in System RPL (see Jim Donnelly's book for details), or you can use the library at <http://www.hpcalc.org/programming/misc/infm2v1.zip>

I figured that the System RPL version of DoInputForm was unpleasant enough to use on a casual basis so I wrote a simple library that lets you use check boxes and choose lists from User RPL.

All it does is stop execution of INFORM just before DoInputForm, then search the stack and change object types to check and choose objects. Then it restarts the INFORM command. At the end, it tidies up TRUE/FALSE to 1/0 etc...

Example syntax:

```
"Test" {
"A" "B" "C" } 1 {} { :CHECK: 1  :CHOOSE: { 1 2 3 }  NOVAL }
IF INFORM2 THEN
   ...insert code here...
END
```

8.22 What is the syntax for the INPUT command?

From: John Meyers

INPUT allows you to display a message and then edit a given string using the command line editor; the edited string is its result.

Arguments:

Level 2:

"message"

The message is displayed in "medium font" in the upper part of the "stack display area" (the stack itself is not visible during the INPUT command). The message may be an empty string, or may contain newlines for a multiple-line message.

Level 1:

The most general format is a non-empty list containing up to three optional specifications, which may appear in any order:

{ "string" { row column } mode(s) }

"String" is the initial text which will appear in the command line; it may be an empty string, or may contain newlines for a multiple- line text. If you do not need to specify any other of the optional arguments, you may omit the list itself and supply only the string on level 1. If the string is omitted from an argument list, the default is an empty string.

Mode(s) are zero or more of the following special reserved names:

- Type a Greek Alpha symbol (Alpha right-shift A) to start the editor with Alpha keyboard mode on.

  You might select this mode if you are inviting the user to type general text; you would not type quotation marks around the entered text, because INPUT automatically returns as a string all the characters that are typed into the edit line.

- The word ALG starts the editor in Algebraic entry mode (ALG annunciator on); PRG mode is always on by default.

- The letter V will check the syntax of the entire command line when ENTER is finally pressed, in exactly the same way that the command line editor normally does, disallowing an exit if there is an RPL syntax error. However, unlike ordinary editing in the command line, the command line text is still returned as a string, and is not compiled for you (you can do OBJ-> yourself afterwards if you want to compile the string).

  For example, if you are expecting two numeric values to be entered, and if you enter 123 456, INPUT returns "123 456"; if you then do OBJ-> on that result, you will finally get 123 and 456 as separate values on the stack.

  Row and Column optionally specify where the cursor will initially appear (default is at the end of the entire string), and whether the cursor will start in "insert" or "replace" mode (default is insert).

  Row numbers start at 1 for the topmost row; column numbers start at 1 for the first character in a row. Row zero means the bottom row; column zero means the end of a row.

  Instead of { row column } you may also simply count character positions within the original "string" argument, supplying just one number instead of a list of two numbers.

  To cause the cursor to be in "replace" mode rather than "insert" mode, the first number specifying cursor position should be made negative.

If the user presses CANCEL during INPUT when there is some text in the command line, all the text will be erased without canceling INPUT; if CANCEL is pressed again (or with no text), then INPUT is canceled.

Examples of use in a program:

```
"Enter A, B, C" {
":A:
:B:
:C:" { 1 0 } V } INPUT
```

Note: Put newlines between the three lines of the initial string.

When executed, the screen shows:

```
Enter A, B, C
:A:(cursor appears here)
:B:
:C:
```

If you now press 1 DownArrow 2 DownArrow 3 ENTER, the string returned on the stack is:

```
":A:1
:B:2
:C:3"
```

If you then do OBJ-> on this string, you get (on separate stack levels):

```
:A: 1
:B: 2
:C: 3
```

The optional "tags" do not interfere with using the numeric values in functions, or storing them (tags are automatically removed in each case).

Another example:

```
"Type a new notecard" { Alpha-symbol } INPUT
```

This allows the user to type any free-form text, which is returned as a string upon pressing ENTER.

Simplest possible example:

```
"" "" INPUT  (no message, empty initial command line)
```

Potential problems with INPUT (if OBJ-> is to be used afterwards to get the values entered):

- You can't force the user to enter the exact number of values expected, so you may want to program defensively (check stack depth, etc...)
- Entries such as command names (e.g. CLVAR), program names, etc... will get executed by a subsequent OBJ->, producing undesirable effects (methods for dealing with this problem are available).

For applications expecting non-text values to be entered, the HP48G/GX INFORM command is more immune to these problems, and affords much tighter control over what is entered, in addition to providing a more beautified display form, a separate prompt for each input field, etc. The syntax for INFORM is detailed elsewhere in the FAQ.

9 Appendix A: Various Useful Functions

9.1 ASC Functions

Note: Although this document mentions SX only, ASC\-> and \->ASC work on both the SX and GX.

From: Bill Wickes

ASCII Encoding HP48 SX Objects

Sending an HP48 SX object via electronic mail can be difficult if the object does not have an ASCII form, such as is the case for library objects. There are various encoding schemes available on different computer systems, but these require that the sender and receiver have similar computers, or at least compatible encode/decode schemes. The programs listed below perform the encoding and decoding on the HP48 SX itself, which has the advantage of being completely independent of any computer.

The programs are nominally called \->ASC and ASC\->. The former takes an object from the stack and converts it to a string, in which each nibble of the object and its checksum is converted to a character 0-9 or A-F. (The object must be in RAM, otherwise a "ROM Object" error is returned.) For sake of easy inclusion in e-mail letters, the string is broken up by linefeed characters after every 64 characters.

ASC\-> is the inverse of \->ASC: it takes a string created by \->ASC and converts it back into an object. When you transmit the encoded strings, be sure not to change the string; ASC\-> uses the checksum encoded in the string to verify that the decoding is correct. An "Invalid String" error is returned if the result object does not match the original object encoded by \->ASC. When you upload a string to your computer, use HP48 translate mode 3 so that the HP48 will convert any CR/LF's back to LF's when the string is later downloaded.

Two versions of ASC\-> are included here. The first (P1) is in HP48 user language, using SYSEVALs to execute system objects. P2 is a string that the setup program uses P1 to decode into an executable ASC\-> - then P1 is discarded. The second version is more compact than the first, and also uneditable and therefore safer (but it can't be transmitted in ASCII form, which helps to make the point of this exercise).

Here are the programs, contained in a directory (if you have problems with this and you are using the HTML version of the FAQ, you may wish to try the plain text version or download a binary copy of the program):

```
%%HP: T(3)A(D)F(.);
DIR
P1              @ ASC\-> Version 1.
\<<
  IF DUP TYPE 2 \=/
  THEN "Not A String" DOERR
  END RCWS \-> ws
  \<< 16 STWS
    #0 NEWOB SWAP DUP SIZE
    IF DUP 4 <
    THEN DROP SWAP DROP "Invalid String" DOERR
    END
    DUP 65 / IP - 4 - # 18CEAh SYSEVAL
    "" OVER # 61C1Ch SYSEVAL
    SWAP # 6641F8000AF02DCCh
    # 130480679BF8CC0h # 518Ah SYSEVAL
```

```
    # 19610313418D7EA4h # 518Ah SYSEVAL
    # 7134147114103123h # 518Ah SYSEVAL
    # 5F6A971131607414h # 518Ah SYSEVAL
    # 12EA1717EA3F130Ch # 518Ah SYSEVAL
    # 280826B3012808F4h # 518Ah SYSEVAL
    # 6B7028080BEE9091h # 518Ah SYSEVAL
    # BE5DC1710610C512h # 518Ah SYSEVAL
    # 705D00003431A078h # 518Ah SYSEVAL
    # 3D8FA26058961431h # 518Ah SYSEVAL
    # 312B0514h # 518Ah SYSEVAL
    # 18F23h SYSEVAL
    DUP BYTES DROP 4 ROLL
    IF ==
    THEN SWAP DROP
    ELSE DROP "Invalid String" DOERR
    END ws STWS
  \>>
\>>

P2      @ ASC\->  Version 2.  To be converted by ASC\-> version 1.

"D9D20D29512BF81D0040D9D20E4A2090000000075660747266365088130119 20
140007FE30B9F060ED3071040CA1304EC3039916D9D2085230B9F06C2A201200
094E66716C696460235472796E676933A1B21300ED30FD5502C230C1C1632230
CCD20FA0008F14660CC8FB97608403104AE7D814313016913213014117414317
4147061311 79A6F5C031F3AE7171AE214F8082103B6280821909EEB0808207B6
215C0160171CD5EB870A13430000D50713416985062AF8D341508813044950B9
F06BBF06EFC36B9F0644230C2A201200094E66716C696460235472796E676933
A1B2130B21300373"

P3      @\->ASC.     To be converted by ASC\->.
"D9D20D2951881304495032230FD5502C230A752688130ADB467FE30322306AC3
0CB916E0E30CBD30F6E30C1C1632230CCD20DC0008F14660CC8FB97608403104
AE7D8143130169174147061741431311534AC6B4415141534946908D9B026155
4A6F53131F3AE731A014C161AE215F08082103A6280821939EEC08082170A621
4C161170CD56B870A18503430000D5071351796A9F8D2D02639916D9D2085230
C2A209100025F4D402F426A6563647933A1B2130A2116B213033C0"

SETUP   @Automatic setup program
\<< P2 P1 'ASC\->' STO
    P3 ASC\-> '\->ASC' STO
    { P1 P2 P3 SETUP } PURGE
\>>

END
```

Installation instructions:

1. Save the above text into a text file named CONV (for example). Be sure that you leave the strings exactly as entered above, with no extra spaces or other invisible characters at the beginnings or ends of the lines.

2. Set the HP48 SX into ASCII transfer mode.
3. Using Kermit, download CONV text file to the 48, verify its checksum (6C8Ah).
4. Execute CONV to make it the current directory.
5. Execute SETUP.
6. The directory CONV now contains ASC\-> and \->ASC, ready to use.

To archive the decoded versions of ASC\-> and \->ASC back on your computer, be sure to set the HP48 SX in binary transfer mode before uploading.

Disclaimers:

- Use the programs at your own risk. Any time you delve into the SYSEVAL world, there are increased dangers. Archive your 48 memory before experimenting with these programs! Also, verify the checksums of objects defined above to make sure they have been downloaded correctly, before executing ASC\->.
- I will not answer questions about how the programs work. This is not because of any great secrecy, but rather because it's hard to give any answer that doesn't lead to more questions, and more, and more...
- 48 hackers are welcome to mine any nuggets they can from the programs, and from the fact that \->ASC is a convenient way to decompile an object.

9.2 OBJFIX

When a binary object received by Kermit on the HP-48 is left as a string beginning with HPHP48, OBJFIX will extract the HP-48 object if the only problem is that extra bytes got appended to the end.

OBJFIX takes a variable name in stack level 1 and modifies the contents of the variable if no other problems are detected.

Note: This is like FIXIT by Horn and Heiskanen on Goodies Disk 8, but this one is by HP and so I suppose it's more reliable. Although it fails the test cases included with FIXIT, that may be because they were artificially contrived cases. Try both on real-world downloads that need fixing. Which do you like better?

OBJFIX.ASC

```
%%HP: T(3)A(D)F(.);
"D9D202BA81D9F81B2040D9D20F2A26DA91629C8145126489162C23072C80CCD2
0BD0008FB97601471081341641420C2818F24D534501008B2F41643150D73B840
58405438314A161966D2BF6BF6A6F5BE16314213114334CF8208A6F58F235A04
55136D7D4EA494D231A1CA101110131CA130DBE284F8FC0760D41198F29960D4
130142119EA1408F5E0108D341503223072D70B2130B21301460"
```

9.3 FIXIT

From: Joe Horn and Mika Heiskanen

PURPOSE:

Converts a badly uploaded string into the original object.

THEORY:

A lot of folks upload HP48 objects poorly, such that when you download them, you just get strings full of garbage that look something like this:

```
"HPHP48-E#c&r$a%p@!*!..."     [looks familiar, eh?]
```

That's because they uploaded it using XMODEM, or managed to screw it up some other way. The following FIXIT program takes such a string and extracts the actual HP48 object that they originally intended to upload (if at all possible).

Such object extraction can be done by hand, but it's too dangerous. FIXIT minimizes the danger of Memory Clear. It checks whether the extracted object is a valid one, and if not, drops it from the stack before the HP48 attempts to display it. All of the many bad downloads I've archived over the years are fixed by FIXIT, whereas about half of them cause a Memory Clear when extracted manually. No guarantees, however. Use at your own risk.

The actual extraction is done by a "Code object" written by Mika Heiskanen. The User RPL "shell" around this code object is what minimizes the danger of Memory Clear; it was written by Joe Horn.

INSTRUCTIONS:

BACKUP YOUR MEMORY, just in case the string contains a logic bomb.

Place the bad download on the stack (see "HPHP48-...") and run FIXIT.

Possible results:

- No error: the object was extracted successfully and is on level 1.
- "Bad Argument Type" error: you didn't have a string on level 1.
- "Bad Argument Value" error: the string wasn't of the proper form; it must be an "HPHP48-..." downloaded string.
- "Invalid Definition" error: the object was mangled in transmission so badly that its end was lost; the object cannot be extracted.
- "Undefined Result" error: there is no HP48 object in the string.
- "Recover Memory? YES/NO": the string contained a bomb, and FIXIT detonated it. Press YES to sift through the shrapnel and rubble in a feeble attempt to resurrect the dead. Press NO to bury them.

EXAMPLES:

To do the following examples, download the FIXIT directory to your HP48 and get into it.

- Press HI. See "HPHP48-E...", a badly uploaded download. Before pressing FIXIT to fix it, try doing what we all used to do: press EDIT to see if we can recognize anything (usually a futile attempt). We see:

  "HPHP48-E... << Melancholy Baby >>"

  But looks can be deceiving; press ON to exit the editor, and then press FIXIT to extract the intended upload:

  << Happy Camper >>

- Press WTAV; see another garbage download. But EDIT refuses; the string contains null characters. Press FIXIT; see successfully extracted directory.
- Press BAD1. Notice that it looks exactly like WTAV. (Press WTAV, compare, then DROP). But its ending is all messed up; manually extracting WTAV from BAD1 can cause Memory Clear. Press FIXIT and see "Error: Invalid Definition" indicating that the object inside BAD1 is so mangled that its end cannot be located.
- Press BAD2. Looks like WTAV again. But its body is messed up; manually extracting it would create an External object that could cause Memory Clear if evaluated. Press FIXIT and see "Error: Undefined Result" indicating that there is nothing recognizable inside BAD2.

FIXIT.ASC

```
%%HP: T(3)A(D)F(.);
"69A20FF7CE2000000040241444234OC2A203B000840584054383D25403A20FF7
25000000004035359474400D9D20E16329C2A2DBBF13013216DF1406A1C42328D
BF193632B213034000407545146540D9D20E163292CF1EFFB1DBBF1EBFB150FA
193632B213000303030303400040241444134OC2A203B000840584054383D25
469A20FF725000000004035359474400D9D20E16329C2A2DBBF13013216DF1406
A1C42328DBF193632B213034000407545146540D9D20E163292CF1EFFB1DBBF1
EBFB150FA193632B2131313131313134C000407545146540C2A203B000840584
054383D25469A20FF725000000004035359474400D9D20E16329C2A2DBBF13013
216DF1406A1C42328DBF193632B213034000407545146540D9D20E163292CF1E
FFB1DBBF1EBFB150FA193632B213000303030303400020849420C2A20570008
40584054383D254D9D20E163284E2050841607079784E20603416D6075627936
32B2130A0BA02D456C616E63686F6C697022416269702BB28000506494859445
50D9D20E16323CE2278BF168BC1ED2A2167E1AFE22D9D203CE2278BF19C2A274
3A2C58C1C2A203100084058405438 3D2167E1AFE22D9D2078BF18B9C1DBBF1AA
F028DBF1CCD201200014713717917F137145142164808C5BF22D9D2033920200
000000005150933A1B21305DF22B21305BF22D9D203392020000000000004150
933A1B21305DF223CE2278BF168BC1D8DC1167E1AFE22D9D203FBF1339202000
000000027709 33A1B21305DF223CE2278BF19D1A1DBBF18DBF1E0CF1D5CE1AF
E22D9D208DBF133920200000000000030933A1B21305DF22CB2A193632B2130
B21303D4F"
```

9.4 LASTX

The LASTX function is useful in calculations where a number occurs more than once. By recovering a number using LASTX, you do not have to key that number into the calculator again. Note however that LASTX uses the built in last argument feature, so if you use LASTX you will lose the contents of your LASTARG.

For example, calculate:

```
 96.704 + 52.394706
--------------------
      52.394706

Keystrokes:                    Stack:
-----------------              --------------------
```

```
96.704 ENTER                    96.704

52.304706 +                     149.098706

LASTX                           149.098706
                                52.304706

/                               2.84568265351
```

```
@ This is a version of LASTX for the HP48
@
%%HP: T(3)A(D)F(.);
\<< DEPTH \-> n
  \<< LASTARG DEPTH n
- DUP \-> s
    \<< ROLLD s 1 -
DROPN
    \>>
  \>>
\>>
```

9.5 Compact Data Storage

From: Jim Donnelly

A simple length-encoding technique can be put to use for a free-format, very compact multi-field data storage system. Two tiny programs, SUBNUM and STRCON are here to help the process, and are listed near the end of this note. At the end of the note is a directory that may be downloaded into the HP48 that contains the examples.

The principle is to store starting indices in the beginning of a string that point to fields stored subsequently in the string. The indices are stored in field order, with an additional index at the end to accommodate the last field. There are several small points worth mentioning:

- Fields may be 0-length using this technique.
- The execution time is uniform across all fields.
- This technique saves about 4 bytes per field after the first field, because the string prologue and length are omitted for fields 2 -> n.

EXAMPLE:

```
                Indices  |          Fields
Character                |     1 11111111 12222222222
Position :   1  2  3  4  |567890 12345678 90123456789
             +--+--+--+--+------+--------+-----------+
String :     | 5|11|19|30|Field1| Field2 |  Field 3  |
             +--+--+--+--+------+--------+-----------+
```

This is a string that contains 3 fields, and therefore 4 index entries. The first field begins at character 5, the second field begins at character 11, and the third field begins at character 19. To keep the

pattern consistent, notice that the index for field 4 is 30, which is one more than the length of the 29 character data string.

To extract the second field, place the string on the stack, use SUBNUM on character 2 to extract the starting position, use SUBNUM on character 3 to extract the (ending position +1), subtract 1 from the (ending position+1), then do a SUB to get the field data. NOTE: The index for field 1 is stored as char. data 5, NOT the string "5"! To place the field index for field 1 in the string, you would execute "data" 1 5 CHR REPL.

PROGRAM:

The following program accepts an encoded data string in level 2 and a field number in level 1:

```
DECODE   "data"  field#  -->  "field"

<<  --> f
  <<
    DUP f SUBNUM                ; "data" start -->
    OVER f 1 + SUBNUM           ; "data" start end+1 -->
    1 -                         ; "data" start end -->
    SUB                         ; "field" -->
  >>
>>
```

DATA ENCODING:

The following program expects a series of 'n' strings on the stack and encodes them into a data string suitable for reading by the first example above.

The programs SUBNUM and STRCON are used to assemble the indices.

```
ENCODE      field n  ...  field 1   n   -->  "data"

<< DUP 2 + DUP 1 - STRCON --> n  data
  <<
    1 n
    FOR i
      data i SUBNUM OVER SIZE   ; ... field index fieldsize
      + data SWAP               ; ... field "data" index'
      i 1 + SWAP CHR REPL       ; ... field "data"'
      SWAP + 'data' STO         ; ...
    NEXT
    data                        ; "data"
  >>
>>
```

In this example, four strings are encoded:

```
Input:  5: "String"
        4: "Str"
        3: "STR"
        2: "STRING"
        1:         4
```

Output: "xxxxxSTRINGSTRStrString" (23 character string) (The first five characters have codes 6, 12, 15, 18, and 24)

VARIATION:

The technique above has a practical limit of storing up to 254 characters of data in a string. To overcome this, just allocate two bytes for each field position. The code to extract the starting index for becomes a little more busy. In this case, the index is stored as two characters in hex.

```
               Indices  |          Fields
Character               | 11111 11111222 22222223333
Position :   12 34 56 78|901234 56789012 34567890123
            +--+--+--+--+------+--------+-----------+
String :    |09|0F|17|21|Field1| Field2 |  Field 3  |
            +--+--+--+--+------+--------+-----------+

<<  --> f
  <<
     DUP f 2 * 1 -           ; "data" "data" indx1 -->
     SUBNUM 16 *             ; "data" 16*start_left_byte  -->
     OVER f 2 * SUBNUM +     ; "data" start
     OVER f 2 * 1 + SUBNUM   ; "data" start end_left_byte -->
     16 * 3PICK f 1 + 2 *
     SUBNUM + 1 -            ; "data" start end -->
     SUB                     ; "field"  -->
  >>
>>
```

TWO VERY TINY HELPFUL PROGRAMS:

```
SUBNUM          "string"  position  -->  code

<< DUP SUB NUM >>

STRCON          code  count  -->  "repeated string"

<< -->  code count
  << "" code CHR 'code' STO
     1 count START code + NEXT
  >>
>>
```

Alternative Solution to the Problem:

From: Matjaz Vencelj *<vencelj@fmf.uni-lj.si>*

Jim allocates two bytes for each index entry (to handle longer strings), but on the other side obviously only uses values 00...FF for index dublets, which just doesn't make sense. He's at a 16*16 = 255 chars limit again!

I have put together a working set of commands which support up to 65K strings, using two-byte indexing.

The encoder (Encode) is User-RPL program which calls the binary N2C which converts the level 1 real into a 2 character string:

```
@*** Encode ***
%%HP: T(3)A(R)F(.);
```

```
\<<
  IF DUP TYPE 0 = THEN 514 DOERR END
  \-> N
    \<<
    N 1 + 2 * #18CEAh SYSEVAL #45676h SYSEVAL
    1 N FOR I I 2 * 1 - OVER SIZE 1 + N2C REPL SWAP + NEXT
    N 2 * 1 + OVER SIZE 1 + N2C REPL
    \>>
\>>

@*** N2C, cksum=#8919h ***
%%HP: T(3)A(R)F(.);
"D9D202BA812BF819FF30D9D20AEC8111920001007FE3057A50C574632230 57A5
0EE250B2130B21307206"
```

ASCII download them with translate code 3, then call program 'Encode' with data strings in levels 2..n+1 and a real n in level 1 indicating the number of strings.

The string decoder (Decode), which is usually speed-critical, is the following \->ASC encoded binary:

```
@*** Decode, cksum=#38E1h ***
%%HP: T(3)A(R)F(.);
"D9D20D8A81D9F811192013000D9D20AEC8113D26CA130F6E30CA130E0E305080
311920001002CE30CAF0650803CBD30CAF06FED30F6E30CA130E0E3050803119
20001002CE30CAF0650803CBD30E0E3033750B2130B21309534"
```

It takes a 'database' string on level 2 and a real (record position) on level 1, then comes back with the appropriate substring (record).

9.6 HP82240B Printer Codes

From: Jarno Peschier

Size of physical row

One printed row is either 24 normal characters, 12 expanded characters or 168 pixels wide. This means that one normal character has a width of 7 pixels. Any printed data "falling off the row" will be truncated and ignored by the printer.

Reset

```
ESC 255d
```

This resets the printer to the following state: `Roman8` character set (watch out: the power-up character set is `ECMA94`), both expanded and underline printing off, buffer cleared.

Self test

```
ESC 254d
```

This causes the printer to print a selftest pattern. This mainly consists of a printout of the `Roman8` character set.

Expanded printing

```
ESC 253d
```

This turns expanded printing on. This means that from this code on all characters that are printed will be printed at twice the normal character width because each column of pixels is printed twice. Has no effect if expanded printing is already on.

```
ESC 252d
```

This turns expanded printing off. This means that from this code on all characters that are printed at normal character width again. Has no effect if expanded printing is already off.

Underlined printing

```
ESC 251d
```

This turns underlined printing on. This means that from this code on all characters that are printed will be underlined because the bottom-most pixel in each columns of pixels is now always on. Has no effect if underlined printing is already on.

```
ESC 250d
```

This turns underlined printing off. This means that from this code on all characters that are printed will not be underlined anymore. The bottom-most pixel in each columns of pixels is printed as it is defined for the printed character. Has no effect if underlined printing is already off.

Character sets

```
ESC 249d
```

This switches the printer to use the `ECMA94` character set. This set is 100% identical to the character set used in HP48 calculators. This is the power up default of the printer. Has no effect if `ECMA94` is the current character set already.

```
ESC 248d
```

This switches the printer to use the `Roman8` character set. This set is different than the character set used in HP48 calculators in the 128 last characters. As far as I know this set is used by older printers like the HP82240A and by HP28 calculators (hence the need for the `OLDPRT` command in HP48 calculators if you are printing to a HP82240A). This character set is turned on if you reset the printer with the reset code. Has no effect if `Roman8` is the current character set already.

Graphics

```
ESC n data (with n between 1 and 247)
```

This causes the printer to print graphics specified by the specified data (one byte per pixel column). The value n specifies the number of bytes of data that follow the printer code that will be interpreted as graphics data. Any data after pixel column 168 will be truncated and ignored.

10 Appendix B: GX Specific Information

10.1 What's new in the HP48 G/GX?

From: Joe Horn

AUTOMATIC LIST PROCESSING

Almost all commands that did not accept list(s) as their arguments can do so now. Here are just a few examples:

```
{ 1 2 3 } SF sets flags 1, 2, and 3
{ 1 2 3 } SQ  -->  { 1 4 9 }
{ 2 4 } 10 /  -->  { .2 .4 }
10 { 2 4 } /  -->  { 5 2.5 }
{ 10 12 } { 2 4 } /  -->  { 5 3 }
{ .1 .2 .5 } ->Q  -->  { '1/10' '1/5' '1/2' }
{ freq freq ...} { dur dur ... } BEEP can play a song with no
  audible hiccup between tones.
```

Since + has always been used to concatenate lists, a new ADD function exists to add the elements of two lists, like this:

```
{ 1 2 3 } { 4 5 6 } ADD returns { 5 7 9 }, whereas
{ 1 2 3 } { 4 5 6 }  +  returns { 1 2 3 4 5 6 } as it did
  before.
```

The only commands which do not have automatic list processing are:

- those which never get a Bad Argument Type error (like DUP),
- meta-object commands (like ROLL),
- program branch structures (like FOR), and
- commands that specifically work on lists (like GET).

Sometimes the results are non-obvious, for example:

```
5  { A B C }  STO           -->  A=5, B=5, C=5
{ 5 6 7 }  'A'  STO         -->  A={ 5 6 7 }  (same as on SX)
{ 5 6 7 }  { A B C }  STO   -->  A=5, B=6, C=7
```

List processing is only recursive for ->Q and ->Qpi.

PORTS AND MEMORY

The HP48 G, like the 48 S, only has 32K RAM. The GX, unlike the SX, has 128K RAM built-in. Card slot 1 can contain another 128K (maximum), but card slot 2 can contain up to 4 megabytes of RAM.

Only port 1 can be merged in the GX. Card slot 2, which is intended for large-capacity RAM cards, is permanently "free", and is automatically divided up into 128K "ports", each of which becomes Port 2, Port 3, Port 4, etc. Up to 4 Megabytes can be plugged into slot 2, which would then become Port 2 through Port 33. (Although the FREE and MERGE commands were kept for HP48 SX compatibility, GX users will prefer the new FREE1 and MERGE1 commands). Therefore the maximum amount of merged main memory is 256K (unlike the SX which allowed up to 288K) after MERGE1; the maximum amount of fully online free independent memory is 4224K after FREE1.

LOCAL VARIABLES

Variable names prefixed with a <- (backarrow character) are compiled as local (temporary) variable name objects even if they're not explicitly after FOR or ->. This allows programs to share values through local variables, which is much faster than sharing values through global variables, and they get purged automatically.

SPEED

CPU clock speed is double the S/SX's, but throughput is estimated to be only 40% faster, primarily due to the fact that all RAM & ROM is now bankswitched (on the S/SX only a 32K portion of the ROM required bank switching), and it still has the same 4-bit bus bottleneck.

IMPROVED COMMANDS:

- AXES can now also specify the spacing of the tick marks.
- DEPND can now also specify the initial values and tolerance for the new DIFFEQ plot type.
- REPL and SUB now work on arrays.

HP SOLVE EQUATION LIBRARY CARD COMMANDS:

- AMORT, amortization calculations
- CONLIB, starts Constants Library catalog
- CONST, returns value of a named CONLIB constant
- DARCY, calculates Darcy friction factor
- EQNLIB, starts Equation Library catalog
- F0lambda, calculates black-body power fraction
- FANNING, calculates Fanning friction factor
- LIBEVAL is a generalized form of the EQ card's ELSYSEVAL; it executes any XLIB by its library number
- MCALC, marks an MSOLVR variable as "not user-defined"
- MINEHUNT, starts the "Minehunt" video game
- MINIT, initializes Mpar from 'EQ' for MSOLVR
- MITM, customizes title & menu of MSOLVR's screen
- MROOT, solve for variable(s) in MSOLVR
- MSOLVR, shows Multiple Equation Solver menu
- MUSER, marks an MSOLVR variable as "user-defined"
- SIDENS, density of silicon as function of temperature
- SOLVEQN, starts solver for specified EqLib equation(s)
- TDELTA, subtracts temperatures like "-" ought to but doesn't
- TINC, adds temperatures like "+" ought to but doesn't
- TVM, shows the financial calculator (Time Value of Money) menu
- TVMBEG, sets payments-at-beginning-of-periods mode
- TVMEND, sets payments-at-end-of-periods mode

- TVMROOT, solves for a TVM variable
- ZFACTOR, calculates gas compressibility factor Z

Note: The EQ Card's Periodic Table and Tetris game are not in the HP48 G/GX, but the EQ Card can be used in the GX if those applications are needed. Tetris was not included because no agreement on royalty was reached. The Periodic Table is available separately as freeware on HPCVBBS.

NEW ARRAY COMMANDS:

- COL+, inserts a column vector into a matrix or a number into a vector (like INSCOL/PUTCOL in Donnelly's Tool Library)
- COL-, deletes a column from a matrix or number from a vector (identical to DELCOL in Donnelly's Tool Library)
- COL->, combines multiple column vectors into a matrix
- ->COL, breaks a matrix into multiple column vectors (like repeated GETCOL in Donnelly's Tool Library)
- COND, column norm condition number of a square matrix
- CSWP, swaps two columns in a matrix (like EXCOL in Donnelly's Tool Library)
- ->DIAG, returns vector of major diagonal elements of a matrix
- DIAG->, creates matrix with specified diagonal elements
- EGV, eigenvalues and right eigenvectors of a square matrix
- EGVL, eigenvalues of a square matrix
- FFT, discrete Fourier transform
- IFFT, inverse discrete Fourier transform
- LQ, returns the LQ factorization of a matrix
- LSQ, minimum norm least-squares solution to an ill-determined system of linear equations
- LU, returns the Crout LU decomposition of a square matrix
- PCOEF, returns polynomial with given roots (inverse of PROOT)
- PEVAL, evaluates polynomial at x
- PROOT, finds all roots of polynomial (inverse of PCOEF)
- QR, returns QR factorization of a matrix
- RANK, rank of a rectangular matrix (uses flag -54)
- RANM, creates matrix with random elements
- RCI, multiplies elements in one row of a matrix by a scalar
- RCIJ, does RCI then adds the result to a row
- ROW+, inserts a row vector into a matrix or a number into a vector (like INSROW/PUTROW in Donnelly's Tool Library)
- ROW-, deletes a row from a matrix or number from a vector (identical to DELROW in Donnelly's Tool Library)
- ROW->, combines multiple row vectors into a matrix

- ->ROW, breaks a matrix into multiple row vectors (like repeated GETROW in Donnelly's Tool Library)
- RSWP, swaps two rows in a matrix (identical to EXROW in Donnelly's Tool Library)
- SCHUR, computes the Schur decomposition of a square matrix
- SNRM, spectral norm of an array
- SRAD, spectral radius of a square matrix
- SVD, singular value decomposition of a matrix
- SVL, computes the singular values of a matrix
- TRACE, sum of diagonal elements of a square matrix

GRAPHICS and PLOTTING COMMANDS:

- ANIMATE, displays grobs on the stack sequentially. You can use the defaults, or specify your own delay between frames (can be very fast), the number of times to repeat the sequence, and even the pixel coordinates. It's just like a ROLL REPL loop... except very fast. Note: Charlie Patton converted 17 seconds of the Apollo moon-walk video into HP48 GROBs and ran them with ANIMATE, and it looked very good!
- ATICK, specifies tick spacing on plot axes
- EYEPT, specifies the eye-point coordinates in a perspective plot
- GRIDMAP, selects the new "gridmap" plot type
- PARSURFACE, selects the new "parametric surface" plot type
- PCONTOUR, selects the new "pcontour" plot type
- PICTURE, same as GRAPH command
- SLOPEFIELD, selects the new "slopefield" plot type
- WIREFRAME, selects the new "wireframe" plot type
- XVOL, sets the width of the 3D plotting volume
- XXRNG, sets the width of the 3D target mapping range for gridmap and parametric surface plots
- YSLICE, selects the new "yslice" plot type
- YVOL, sets the depth of the 3D plotting volume
- YYRNG, sets the depth of the 3D target mapping range for gridmap and parametric surface plots
- ZVOL, sets the height of the 3D plotting volume

USER-INTERFACE COMMANDS:

- CHOOSE, displays a point-and-click menu "dialog box"
- INFORM, formatted multi-line input with named fields (nice!!)
- MSGBOX, displays text in a centred box with shadow, then WAITs
- NOVAL, placeholder for unspecified values in INFORM argument list

LIST PROCESSING COMMANDS:

- ADD, adds lists element-wise (see section above)

- DOLIST, evals an object on multiple lists
- DOSUBS, evals a program or command taking arguments from a list
- ENDSUBS, returns the number of loops the current DOSUBS will do
- HEAD, first element in a list or first char in a string (identical to CAR in Donnelly's Tool Library)
- DeltaLIST, list of first finite differences of list objects
- SigmaLIST, sum of the elements in a list
- PiLIST, product of the elements in a list
- NSUB, returns the current list pointer value during a DOSUBS
- REVLIST, reverses the order of the objects in a list (like REVERSE in Donnelly's Tool Library)
- SEQ, list of results from repeated execution of an object (like a FOR/STEP loop but the results go into a list)
- SORT, sorts elements in a list into ascending order, or sorts a list of lists using each list's first element as the key (can be done with LSORT/QSORT in Donnelly's Tool Library)
- STREAM, executes an object on first two elements of a list, then again on the result and the 3rd element, etc. Allows easy creation of things similar to SigmaLIST and PiList.
- TAIL, returns a decapitated list or string (see HEAD above) (identical to CDR in Donnelly's Tool Library)

SYSTEM COMMANDS:

- CLTEACH, clears the 'EXAMPLES' directory created by TEACH
- CYLIN, sets polar/cylindrical coordinate mode
- FREE1, like 1 FREE (see section above)
- MERGE1, like 1 MERGE (see section above)
- PINIT, port initialize, esp. important for 4-Meg RAM card users
- RECT, sets rectangular coordinate mode
- SPHERE, sets polar/spherical coordinate mode
- TEACH, loads the Owner's Manual examples into a dir in HOME
- VERSION, returns the operating system ROM version string and a copyright notice, like this:

```
2: "Version HP48-R"      <-- means version "R"
1: "Copyright HP 1993"
```

- XRECV, X-Modem protocol receive (binary mode only)
- XSEND, X-Modem protocol send (binary mode only)

MATH COMMANDS:

- LININ, tests whether an equation is linear in a given variable
- NDIST, normal probability density
- PCOV, population covariance of SigmaDAT
- PSDEV, population standard deviation of SigmaDAT
- PVAR, population variance of SigmaDAT

- RKF, solves initial value problem using Runge-Kutta-Fehlberg
- RKFERR, change in solution and absolute error using RKF
- RKFSTEP, next solution step with given error tolerance using RKF
- RRK, solves initial value problem using Rosenbrock & RKF
- RRKSTEP, next solution step with given error tolerance using RRK
- RSBERR, change in solution and absolute error using Rosenbrock

MENU NUMBERS and KEY CODES

Many menu numbers have changed, so software that uses # MENU or # TMENU may not work the same as in the HP48 S/SX. (Specifically, only menu numbers 0-3, 28, 30, and 42-59 are the same). Likewise, almost all of the shifted keycodes correspond to new commands and menus, which programmers must take into account; for example, the "RAD" key on the S/SX had the keycode 82.2, but it's 21.2 on the G/GX. The left-shift key, which was orange on the S/SX, is now purple [officially "lavender"], and the right-shift key which was blue on the S/SX is now green [officially "teal"] on the G/GX. Also, the digit-key menus can be activated by both shift keys; left-shift gives the softkey menus like in the S/SX, but the right-shift gives the new user-friendly full-screen menus. The unshifted keys remain identical to the S/SX, except for a cosmetic colour change to match the very dark green of the calculator case.

MANUALS

The G/GX comes with two manuals, a "Quick Start Guide" for beginners, and a cost-cutting, slimmer owner's manual called the "User's Guide" which has only 21 pages about programming, since HP figures that the huge majority of all 48 owners never program it anyway. The power users can buy the optional "Advanced Users Reference Manual" (similar to the S/SX's "Programmer's Reference Manual") which covers programming and the many commands that are not mentioned in the User's Guide. There is no "Quick Reference Guide" like the S/SX came with, although the case still has a pocket for one.

Jim Donnelly has marketed a nice pocket guide, but it's too wide to fit in the case's pocket. The User's Guide is not spiral bound, but is made to open fully and last a long time, since it's not just glued but has sewn signatures like real books, and is printed on quality paper.

Another possibility is "The HP 48G/GX Pocket Guide" by Chris Coffin and Thomas Dick (Grapevine Publications). It's 80 pages long, contains a complete command reference (with input/output stack diagrams), alpha keyboard description, system flag description, as well as examples of how to use various calculator features. This guide is designed to fit nicely in the pocket of the HP case.

FLAGS

Some previously "unused" flags are now used. They are:

```
-14 Clear = end-of-period payment mode (for TVM calculations)
    Set   = beginning-of-period payment mode

-27 Clear = display symbolic complex numbers in coordinate form
              e.g. '(X,Y)'
    Set   = display symbolic complex numbers using 'i'
              e.g. 'X+Y*i'
```

```
-28 Clear = plot multiple equations like the S/SX does (serially)
    Set   = plot multiple equations simultaneously

-29 Clear = include axes in plots (like the S/SX does)
    Set   = omit axes from 2D and statistics plots

-30 is no longer used (it never did anything useful anyhow)

-54 Clear = tiny matrix elements get rounded to zero
    Set   = leaves matrix elements alone
```

The default setting of all these flags is Clear (as in the S/SX).

FLAG BROWSER

There is a System Flag browser which shows the flag number, shows whether it's set or clear, lets you toggle it, and shows in English what the current setting means.

CHARACTER BROWSER

While programming, if you want to type any character at all, press CHARS and a screenful of ASCII characters is displayed that you can browse with the arrow keys, and not only does the screen also show the ASCII code (NUM value) and even the shortcut keyboard key sequence (if any) for each character, but if you press ECHO, it will be "typed" into your program. There's no need any more for the alpha keyboard table.

DIRECTORY MAINTENANCE

Press right-shift VAR to launch a Variable Browser which is a complete memory manager. You can tag multiple objects and copy, move, or delete them all with a single keystroke; there's even a Mark All and an Unmark All, like a real computer. It's slow, however, and has been obsoleted by the very fast PCT library.

FRACTIONAL UNIT POWERS

The S/SX only handled integer powers of units correctly, but the G/GX can use any real number as a unit power.

NAME PLATE

The case has a rectangular indentation in the back like the HP 95LX and 100LX, and it comes with an adhesive metal nameplate that you can get engraved with your name.

XLIB NAMES

All of the new commands in the GX are XLIB names, and therefore take 5.5 bytes in programs. The commands common to the SX and GX take 2.5 bytes each, as they did in the SX.

INPUT FORMS and CHOOSE BOXES

Many operations have two menu types: the old SX style, and a new "drop-down" menu and "input forms" that have the feel of computer dialog boxes. Especially useful for the HP48 beginner.

ENHANCED PRECISION

The internal precision of at least some of the matrix routines has been improved; INV gets better answers on square matrices than the SX did. HP has not released information about which routines were improved, how, and by how much.

IMPROVED DISPLAY

The LCD introduced with revision M of the G/GX is easier to read since it has higher contrast between on/off pixels. It has a slower cycle response time, however, making it difficult to use for rapid-motion video games or any other rapid animation.

10.2 Examples of INFORM, CHOOSE, and MSGBOX

From: Jarno Peschier

Some examples of INFORM, CHOOSE and MSGBOX on the HP48 GX. Just download the entire directory to your calculator and try the programs, change them, modify them and do everything else with them you can think of.

SIMPLE:

This program will demonstrate a simple INFORM input screen with 3 fields (one without type restrictions, one for real or complex numbers and one for strings) with some additional layout. MSGBOX and CHOOSE (with the third parameter equal 0) are used to show what the results of the INFORM command are. The list the INFORM command returns is left on stack so you can see what it looks like.

ANGLE:

This program will demonstrate the use of CHOOSE. It lets you choose between the three possible angle modes (DEG, RAD, GRAD) and when you choose one of them, the corresponding mode is set by evaluating a tiny program containing the right command.

ISOLATE:

This program will demonstrate the use of INFORM in ways that it is used in the calculator itself. It is a very simple shell around the ISOL command (isolation of a variable from an algebraic). It remembers it's settings in a variable called IPAR and the next time the program is run this will be the default values of the INFORM command, so you can isolate for a different variable using the same algebraic you used before, without retyping it. MSGBOX is used for error messages.

TYPELIST:

This program will again demonstrate the use of CHOOSE. It extracts the names of all the internal types of the HP48 GX from ROM and shows them in a CHOOSE-box (alphabetically sorted by name). If you choose one of them, its TYPE number is shown in a MSGBOX.

MATHQUIZ:

This final program will demonstrate the use of INFORM with variable field descriptions and default/reset values. It's will show you 8 fields that are simple math questions for you to solve (addition and subtraction). You can enter all the results and then you will see if your answers were correct (in a MSGBOX). You must fill all the fields. Hint: you can cheat by resetting a field (or all fields).

```
%%HP: T(3)A(D)F(.);
DIR
  SIMPLE
    \<<
      IF
```

```
"AN EXAMPLE OF INFORM"
{ { } { } { } {
"OBJECT:"
"ALL OBJECTS ARE ALLOWED HERE"
} { } { "NUM:"
"ENTER A (COMPLEX) NUMBER"
0 1 } { "NAME:"
"ENTER YOUR FULL NAME"
2 } { } { } } { 3 1
} {
        \<< 440 1
BEEP
        \>> (0,1)
"JARNO PESCHIER" }
{ NOVAL 0 "N.N." }
INFORM
      THEN DUP
"YOU ENTERED:" SWAP
0 CHOOSE DROP
"The list that INFORM produced is still on the stack."
      ELSE
"You cancelled the INFORM."
      END MSGBOX
    \>>
  ANGLE
    \<<
      IF
"ANGLE MEASURE" { {
"Degrees" DEG } {
"Radians" RAD } {
"Grads" GRAD } } 1
CHOOSE
      THEN EVAL
      END
    \>>
  ISOLATE
    \<<
      IF
"A VERY SIMPLE VARIABLE ISOLATOR"
{ { } { "EXPR:"
"ENTER THE EXPRESSION"
9 } { "VARIABLE:"
"ENTER VARIABLE TO ISOLATE"
6 } } { } { }
        IF 'IPAR'
VTYPE 5 \=/
        THEN { }
        ELSE 'IPAR'
RCL
        END INFORM
      THEN DUP 'IPAR' STO
```

```
        IF DUP
NOVAL POS
        THEN DROP
"You must enter an expression and a variable!"
MSGBOX
        ELSE OBJ\->
DROP
          IFERR
ISOL
          THEN
DROP2 "Error: "
ERRM + MSGBOX
          END
        END
      END
    \>>
  TYPELIST
    \<<
      IF
"ALL HP48 TYPES (IN ROM)"
0 27
        FOR msg
          IFERR msg
263 + DOERR
          THEN ERRM
          END msg 2
\->LIST
        NEXT 28
\->LIST SORT 1 CHOOSE
      THEN
"That one has type number "
SWAP + "." + MSGBOX
      END
    \>>
  MATHQUIZ
    \<<
      IF
"A SIMPLE MATH QUIZ"
1 8
        FOR i "'"
RAND 100 * IP +
          IF RAND
0.75 <
          THEN "+"
          ELSE "-"
          END +
RAND 100 * IP + "'"
+ "ENTER RESULT #"
i + 0 3 \->LIST
        NEXT 8
\->LIST DUP
```

```
        \<< \-> X
          \<< X HEAD
2 OVER SIZE 1 - SUB
"=" + X 1 ROT PUT
          \>>
        \>> DOLIST
SWAP
        \<< \-> X
          \<< X HEAD
OBJ\-> EVAL
          \>>
        \>> DOLIST 3
ROLLD { 2 5 } 4
PICK { } INFORM
      THEN
        IF DUP
NOVAL POS
        THEN DROP2
"You didn't fill all the blanks."
        ELSE
          IF SAME
          THEN
"All answers were correct!"
          ELSE
"Not all answers were correct."
          END
        END MSGBOX
      ELSE DROP
      END
    \>>
END
```

10.3 Some useful LIBEVALs

From: Joe Horn

Note well: backup memory before using any of the following! LIBEVAL can clear memory if used incorrectly. Warning to the clueless: LIBEVAL, NOT SYSEVAL!!! If you don't know what a "bint" is, don't use the ones that mention bints.

Example usage: "OUT OF RANGE Try Again" #B0091h LIBEVAL. Try it!

Function	LIBEVAL
Displays message box with grob	#B1000h
CMD last command window	#B2000h
CHARS application	#B2001h
MODES application input form	#B41C1h
flag browser (returns t/f to level 1, just drop it)	#B41CFh
MEMORY application (aka variable browser)	#B41D7h

```
SOLVE application choose box                          | #B4000h
solve equation input form                             | #B4001h
solve difeq input form                                | #B4017h
solve polynomial input form                           | #B402Ch
solve linear systems of equations input form          | #B4033h
solve TVM input form                                  | #B4038h
PLOT input form                                       | #B4045h
SYMBOLIC application choose box                       | #B4113h
integrate input form                                  | #B4114h
differentiate input form                              | #B4122h
Taylor polynomial expansion input form                | #B412Bh
Isolate a variable input form                         | #B412Dh
solve quadratic input form                            | #B4130h
manipulate expression input form                      | #B4131h
TIME application choose box                           | #B4137h
Set alarm input form                                  | #B4138h
Set time and date input form                          | #B415Bh
Alarm browser   (aka alarm catalog)                   | #B416Eh
STAT application choose box                           | #B4175h
single-var stat input form                            | #B4176h
frequencies input form                                | #B417Dh
fit data input form                                   | #B417Fh
summary stat input form                               | #B418Fh
I/O application choose box                            | #B4192h
Send to HP48 input form                               | #B4193h
Print input form                                      | #B4197h
Transfer input form                                   | #B41A8h
Get from HP48  (immediate)                            | #B50FFh
recalls the contents of the reserve variable Mpar     | #E4012h
------------------------------------------------------+----------

LIBEVAL : stack diagram / what it does
-------   ------------------------------------------------------------
#B0091h : $ --> makes a message box with an alert symbol in it
#E0044h : $ --> displays a title line, top center (follow with
          1 FREEZE if you want it to stay there after program ends)
#B2000h : launches the Last Command choose-box
#B2001h : launches the CHARS application; returns nothing if user does
          not press ECHO
#B2002h : launches the CHARS application; returns "" if user does not
          press ECHO
#B41CFh : launches Flag Browser; leaves a True or False on stack, so
          follow this LIBEVAL with a DROP.
#B50A3h : --> current time as hour, min, sec separately, plus an XLIB
          that represents AM/PM/24-hr (just DROP it)
#B50A4h : hh.mmss --> hh mm ss xlib (the xlib represents AM,PM, or
          24-hr mode; just DROP it)
#B50A6h : --> current date as month, day, year (always that order, and
          a two-digit year)
#B50A7h : mm.ddyyyy (or dd.mmyyyy) --> month, day, yr
#B50A9h : #month #yr --> #days_in_that_month (inputs and output are
```

```
          bints; year is two digits, interpreted as between 1991 &
          2090 only)
#B50AAh : #yr --> %0.00yyyy (input is two-digit bint interpreted as
          between 1991 and 2090; output is a real number)
#B50ABh : #mon #day #yr --> #day_of_week (inputs & output are bints;
          year is two digits interpreted as between 1991 & 2090;
          Sunday is #7; if you're in DMY mode, then the input order is
          #day #mon #yr)
#B50B2h : --> { 1 2 3 ... 59 } (not very fast)
#B50B3h : --> { 1 2 3 ... 10 } (very fast)
#B50B4h : --> { 0 1 2 ... 23 } (very fast)
#B50B9h : --> { " 1 January" " 2 February" ... "12 December" }
#B50D5h : --> number of alarms currently set (as a bint)
#E3063h : hxs --> grob (this is an RLL packed-grob uncompressor, used
          by EQ LIB and MINEHUNT; for example, try this:
          #E202Bh LIBEVAL 3 GET #E3063h LIBEVAL PICT STO PICTURE)
#E202Bh : the first of the packed EQ LIB grobs (see above)
#E2069h : the last of the packed EQ LIB grobs (see above)
#E7039h : MINEHUNT packed grob (left screen border)
#E703Ah : MINEHUNT packed grob (right screen border)
#E801Eh : obj --> obj T/F (tests whether object is in temporary memory
          or not; returns System-RPL True or False)
```

11 Appendix C: Details of Bugs

11.1 The EquationWriter Bug

From: Joe Horn

Rev E Behaviour

Clear flag -53 first (the Precedence Flag).

On a Rev E, put '((1+2)/(3+4))^5' on the stack and press down-arrow. You'll see:

```
         5
    / 1+2 \
    | --- |                        (A)
    \ 3+4 /
```

which is as it should be. But now press [orange-shift] [+]; see the message "Implicit () off" momentarily; press [left-arrow] (not backspace), then press the [EXIT] softkey. The expression gets mangled into this:

```
       1+2
    -----------                      (B)
          (5)
     (3+4)
```

which is not equal to expression (A) above! Bug, yes? Press ON to abort the process.

Now set flag -53, then repeat the above procedure. First you see:

```
       5
/ 1+2 \
| --- |                        (C)
\ 3+4 /
```

which is the same as (A) above; but continuing as before, you see:

```
         (5)
/  1+2  \
| ----- |                      (D)
\ (3+4) /
```

which is equal to the original. Thus the bug can be worked around by keeping flag -53 set (not a pleasant solution).

Rev J Behaviour

Major difference: after pressing down-arrow, Rev J goes directly into graphic mode, so you have to press ON and then EXIT to get into the equation editor (which Rev E goes directly into). But that's petty cash compared to the following big change.

The same sequence of operations, first with flag -53 clear, then set, exactly as detailed above, yields these four displays in a Rev J:

```
         5
/ (1+2) \
| ----- |                      (A')
\  3+4  /
```

(notice the extra parentheses?) and then:

```
         5
/ (1+2) \
| ----- |                      (B')
\ (3+4) /
```

which is equal to (A'); nothing at all like expression (B) above! and then:

```
         5
/ (1+2) \
| ----- |                      (C')
\  3+4  /
```

which is the same as (A') above; and then:

```
         5
/ (1+2) \
| ----- |                      (D')
\ (3+4) /
```

which is also equal to (A'). No bug in Rev J.

SUMMARY: Rev A-E have a bug in the EquationWriter that can mangle expressions if flag -53 is clear (the default) and if "Explicit Parentheses" mode is toggled. This bug has been fixed in Rev J.

Unfortunately (as you can see above) Rev J always puts parentheses around polynomial numerators. It is therefore impossible to use the ->GROB command on a Rev J to create a GROB that looks like expression (A) above; the simplest that can be had is expression (A').

Another minor change, while I'm at it: Rev A-E don't change the menu when you press REPL; Rev J automatically flips to the appropriate RULES menu.

11.2 Rotation Rate to Angular Frequency Conversion Bug

About the Bug:

From: Wlodek Mier-Jedrzejowicz <*wacm@doc.ic.ac.uk*>

There is a rotation rate conversion bug in the HP48 G/GX which I have not seen reported here before, so after discussion with the folks at Corvallis I am posting this description. Warning: it is 159 lines long!

First - an example. Put the unit object 60_rpm in level 2 and the unit object 1_r/s in level 1, then execute the command CONVERT. You are asking the HP48 to convert a rotation rate of 60 revolutions per minute into an angular frequency in radians per second. 60 rpm is 1 revolution per second, or 2pi radians per second. No HP48 G/GX will give this answer! Not everyone uses rpm or is even aware of the existence of this unit - it is one of the extra units in the UTILS menu of the Equation Library - so here is a second example - add 2pi radians per second to one Hertz. Put 6.2832_r/s in level 1, 1_Hz in level 1, and add. You are adding an angular frequency of two pi (one cycle) per second to a rotation rate of one per second, so the result should be a frequency of two Hertz. On an HP48 S/SX that is the answer. On an HP48 G/GX it is not.

When units are converted, by CONVERT, or during arithmetic on unit objects, the level 2 object is first turned into "base units", and then the result is converted into the units of the level 1 object. On the HP48 S/SX, the "base unit" of angles is one rotation (or a "unit circle" or a revolution or a cycle). So, the angle unit of rpm (a revolution) or of Hz (a cycle if Hz is treated as a rotation rate) is already in base units - conversions to angles involving rpm and Hz automatically work correctly. On the HP48 G/GX, the "base unit" of angles is the current angle mode (DEG, RAD or GRAD) - so any conversion from rpm or Hz (or any formula which works in cycles, rotations, revolutions, unit circles) to angles should be preceeded by a conversion from the unit circle to the current angle. Apparently no-one noticed this would be necessary, because it all worked automatically on the HP48 S/SX.

So, when you convert 60_rpm to units of _r/s, an HP48 G/GX converts not 60 rotations but 60 "base angle units" per minute to radians/second. In RAD mode, you get 1 radian per second. In DEG mode you get 1 degree per second, and in GRAD mode you get 1 grad per second (in each case expressed in radians). That's three different answers, none of which is correct! Exactly the same happens if you convert 1_Hz to angles per second, and the inverse mistake is made if you convert angles per time to cycles or rotations divided by time.

I first learned of this bug from a member of HPCC (the British club for users of HP handhelds), Peter Embrey. He describes his troubles in articles in the first two 1994 issues of our club journal, DATAFILE (in Volume 13 number 1 pages 12 to 14 and V13n2p6). He was calculating the energy stored by a flywheel - given by the formula (1/2)*I*omega^2 and after a time he decided the answers had to be much too big when he CONVERTed from kg*m^2*(r/s)^2 to W*h on an HP48 GX. It turns out that (r/s) are the correct units to get the right answer, but the GX was converting to degrees per second as it was in DEG mode, so his answer was too large by a factor of (360/2pi)^2 - a factor of about 3,300.

In this case, his HP48 SX was not much better, since it converted from radians to unit circles. The way to get the correct answer is to use an HP48 G or GX in RAD mode - or to divide out the radians from the formula before using CONVERT. This is not yet a bug, but needs as much care as does use of temperature units on the HP48. But when Peter tried to deal with the problem by working in rpm, he came upon the bug described above. My thanks to Peter for putting me on the trail!

Apparently this bug not been reported before - at least my friends in HP tell me that it was not on their list of known problems until I told them of it. (This means it is not fixed in the new revision R.) Why not - does everyone know about it and work around it without thinking to tell anyone else? Or does no-one use their HP48 to do calculations on rotating bodies - or do most people do calculations with rotating bodies in such a way that they do not encounter this problem? Could there be hundreds of students and engineers out there calculating and designing things on their HP48 G/GX and getting wildly inaccurate results? Has anyone built a disk drive or a jet engine which rotates far too fast and will disintegrate because of this? No, of course not, all engineers know that any design calculation absolutely must be repeated on two entirely separate calculators or computer programs! :-| Maybe some students have lost marks in exams because of this though - but please, this is not intended to restart the discussion as to whether calculators should be allowed in exams!

I want to underline again that apparently no-one has reported this before - which must mean that few people have been affected by it. It is therefore not a good reason to throw away your HP48 G/GX or get on a high horse and demand that HP replace your HP48 G/GX - but I think it is important that people be warned so they can take appropriate avoiding action. The rest of this message goes into more detail - if you never worry about rotation calculations then you can safely ignore the rest - though you might find it interesting, so don't stop yet :-)

One way to avoid this would be to add a new unit to the HP48 - call it what you like - the "cycle" or "rotation" or "revolution" or "unit circle". As I wrote above, this is already implied in the HP48 S/SX; to see this on an HP48 S/SX, put 360 degrees in level 1 and execute UBASE - the result is 1, meaning that 360 degrees are equivalent to one base unit of angle measurement, but that there is no named HP48 unit corresponding to this. In contrast, UBASE on an HP48 G/GX considers the base unit of angle measurement to be the radian, even though CONVERT behaves as though the base unit is the current angle mode. There appear to be two different norms for base angle units on the HP48 G/GX!

The whole subject gets very little mention in HP's manuals. In the original HP48 SX manual (two volumes, spiral bound), the section on "Dimensionless Units of Angle" in chapter 13, on page 198, warns the reader about the danger of using dimensionless units and states how angle units and scalars are treated. In the later HP48 S and HP48 SX manual (one volume), the same warning is given in "Converting Dimensionless Units of Angle", on page 13-12. The HP48 G Series User's Manual, in "Converting Angular Units" on page 10-7, says that conversion will interpret a scalar according to the current angle mode setting. (A scalar is a pure number with no units.)

For a detailed description, look in the HP48 S/SX edition of "HP48 Insights Vol II", section 21.4.3. This book is written by Dr Bill Wickes, who was the design team leader of the HP48 SX, and who wrote the "Insights" books largely to provide the sort of explanations and details that get left out of manuals. A good explanation of angle units is exactly the sort of thing one can find there! He explains the pitfalls and unavoidable contradictions of working with angles in the HP48 units system and points out that the HP48 S/SX make the somewhat arbitrary choice of using 2pi as the base unit of angles, thereby making conversions between angles per time and Hertz work correctly.

Maybe no-one on the HP48 G/GX team read this while they were making changes from the HP48

S/SX! Why did they change the base unit at all? Most likely they were trying to deal with another contradiction: the units system lets you add pure numbers to angles, since both are dimensionless. If you add the number 1 in level 2 to the unit object 0_r in level 1 on an HP48 S/SX, the number 1 is treated as 1 base unit, or 2pi radians, and the result is 6.2832_r - but if you take the SIN of the number 1 instead, it is not treated as 2pi, but as 1 unit of the current angle mode. The change made on the HP48 G/GX does resolve this contradiction, but at the cost of introducing the bug described above.

As mentioned, a way to resolve the problems involved would be to add the angle unit "cycle" explicitly to the HP48 units system. Hz would then be treated as cycles per second when used in calculations involving rotations - rpm would be treated as cycles per minute, and conversions would go from cycles to the appropriate angle units. This suggestion was made by Peter Embrey in his articles, and the folks at HP accept that this is a good solution - but they have not implemented it yet. In the meantime, be very, very careful when converting between units of rotation rate and units of angular frequency. I would urge everyone who does not yet have a copy of Insights II to buy one and read the relevant section - maybe that will even entice Bill Wickes into publishing his long-awaited HP48 G/GX version of the book!

I have not yet mentioned solid angles. In principle there should be no problem - on both the HP48 S/SX and the HP48 G/GX the base unit of solid angle is a "unit sphere", or 4pi steradians. On the HP48 S/SX you can add the pure number 1 to 0_sr and get 12.5664_sr (4pi steradians). The HP48 G/GX manuals imply that exactly the same should happen, but on my (version L) HP48 GX this gives the error message "Inconsistent Units". This is yet another undocumented difference between the Series S and Series G but at least it is no bug!

Apologies for making this description so long, I hope most people will agree that a subject like this deserves a careful description! For my next trick - some details on the HP48 Random Number Generator.

Addition Insight:

From: Eric Haas <*EHaas@ix.netcom.com*>

Note: The < symbol below is actually the angle character.

The angular conversion bug is actually in the definition of the rpm unit. If you put 1_rpm on the stack, and type UBASE, you get 1.66666666667E-2_1/s. Notice that there is no angular unit in the definition. If the rpm unit is instead defined as 6_</s, all conversions to and from rpms will work just fine. As an easy work-around, define the unit RPM as 6_</s and use that instead of the built-in unit.

If desired, one could also define the unit HZ as 60_rpm or 360_</s. However, as Hz is sometimes used to describe things other than rotation rates, such a definition would not be appropriate for all circumstances.

12 Appendix D: Hardware Additions

12.1 How to Make a Serial Cable

From: Frank Vorstenbosch <*prompt@xs4all.nl*> Revised by: Andrew Chen

Now that you have your HP, you probably want to tap the tremendous amount of programs out there. But how do you do this? You need an HP to PC link. You can buy one, but they tend to be fairly costly. Or

you can build your own. The process requires the following parts:

Required Parts:

- Soldering iron and flux
- RS232-9 or RS232-25 shield (serial port shield)
- RS232-9 or RS232-25 female connector (serial port connector)
- 4 pin HP connector, with sockets spaced 2 mm (NOT 0.1") apart
- Some copper wire

Of the above list of parts, most are pretty easy to acquire, and you should be able to find them at your local electronics store. However, the 4 pin (i.e. male) HP connector can be a bit harder to find.

If you happen to have a broken floppy drive, hard drive, or CD-ROM audio cable lying around, look inside and see if you can find a connector there that will fit the HP48. Do not use a 0.1" connector, as this will damage the pins on your calculator.

HP Connector

The HP connector part of the cable will be the most frustrating part of the link. The reason is that you will probably not be able to find these at your local electronics store, but you should you should find everything else there. Instead, you can either find the connector at used computer stores or you can create the link yourself.

Easy Way

Because the former is much easier, I will give you some tips on where one can find the connecter. First, check your yellow pages and look under "computer". You will find a lot of stores, but look specifically for "computer repair" or "computer parts". Call these stores asking for a "CD-ROM audio cable". If the store carries these cables, go to the store and ask to see the requested cable.

What you should see is a cable with two ends, each with a 4 socket connector. One end should be spaced a bit larger than the other, and this one will not fit into the HP's pins (don't press too hard, or else you will bend the connector pins). The other side should fit like a charm. Don't be worried about that there are only 3 sockets. This is fine because the empty one is a ground. Once you have this cable, cut off the unusable end, and go to the PC connector section.

Hard Way

If are unable to find a place that carries CD-ROM audio cables, or you prefer to make your make, it is possible to build the connector yourself. What you can do is buy IC sockets from a local electronics store. You can usually find them in packs of 8 or more. These IC sockets will be bound together in a hard plastic shell, which also places at unusable intervals. Therefore, it is necessary to strip the IC sockets out of the hard plastic shell (don't worry too much about damaging them, the IC sockets are fairly durable and you have 8 of them).

Next, solder a 'fork' from thin rigid metal wires, to hold the four IC pins spaced at exactly 2 mm while you glue them together with superglue. Glue a plastic 'handle' to the four IC pins to be able to remove the connector from the HP48. You can also indicate the top side of the connector on this handle.

Note that the hole in the HP48 in which the connector should go is not symmetrical; the pins are nearer to the top of the calculator than the bottom, and you can use this to make it difficult to insert the connector the wrong way up.

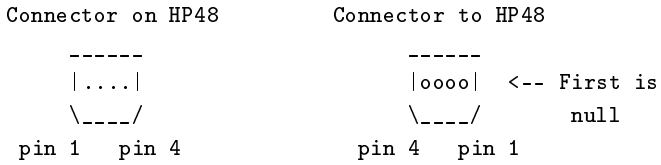
With that done, you can proceed on to the PC side of the connector.

PC Connector

The PC side of the connector is much simpler than the HP side. All you have to do is make a standard serial connector with the parts you bought. However, don't do it yet because you need to solder the wires from the HP connector into the back of the serial connector first.

Making the Connections

Now that you have the two sections done, you can begin making the connections for the actual link. Starting with the HP side, put in the connector and mark the top as "UP". If you have a CD-ROM audio cable with one socket missing, make sure that the empty socket connects to pin 1:



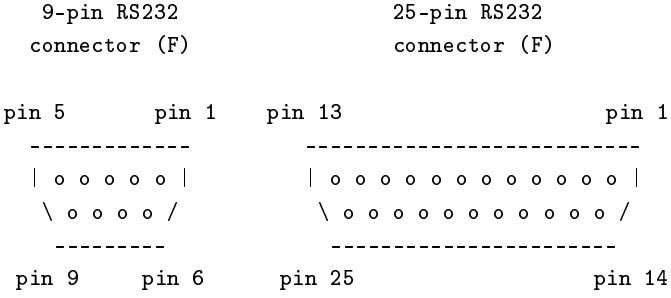
If you made the HP connector the hard way, you have to solder the ends of the IC sockets to long pieces of wire, which will eventually connect to the serial connector on the PC side of the cable.

With these wires done, you must solder the individual wires into the proper places on the PC side. Use this table of pin connections:

HP to PC cable

HP48	RS232-9	RS232-25
1	shield	shield
2	2	3
3	3	2
4	5	7

You can use either a 9 or a 25 pin female sub-D socket for the PC-side of the cable:



Use flexible 4-wire cable to connect the four contacts of your HP48 connector to the PC connector. Pin 1 of the HP48 should be connected to the metal shield of the RS232 connector. Usually it is not easy to solder this shield; first scratching the shield bare (it has some kind of coating) using a screwdriver

or a file will help. If this doesn't work, simply leave pin 1 of the HP48 disconnected. Note that pins 2 and 3 of the RS232 connector must be swapped when you use a 25-pin connector.

Before connecting the completed cable to your HP48, check for short-circuits using an ohmmeter or multimeter set to ohms or "diode test". The HP48 has a built in serial loop back test that can be used to test the serial cable (see the question regarding the ON-KEY combinations).

After you are done, close the shield and connector, and put in all the screws. You should now have a HP<–>PC link, which functions on COM (serial) ports.

Using the Link

With the link finished, it is now ready to be tested. Download some programs (such as those in the Best Program List) that you wish to try. At some point you should go to `<http://www.columbia.edu/kermit/>` to obtain the version of Kermit that suits you best.

However, if you have an alternate communications program (for example, Windows 95 comes with HyperTerminal which you can use) you can delay downloading Kermit. However, it is a highly recommended to obtain because certain programs for the HP48 use special features in Kermit (such as server mode) not available in other communications programs.

With this done, you can begin the actual transfers. Start your communications program, and set the port to COM1 (or whatever your link is plugged into). In Kermit, you would type "SET PORT COM1" and in HyperTerminal you would set the dialog box with the choice of what modem you want to use to "Direct to COM1".

Then change the speeds of the ports to 9600. In Kermit, type "SET SPEED 9600" and in Hyperterminal click on Advanced.

On the HP, go to I/O, and go to Transfer. Set the calculator to Wire, 9600 baud, and Kermit (or X-Modem, if you using it instead). Then, get ready for to receive a file. Note that X-Modem is much faster than Kermit in most situations, especially in long transfers. However, it is not available built-in on the S/SX.

On the PC start sending, and on the HP, start receiving. You should see the transfer arrow on the upper right of the screen on the HP, and it should be flashing. On the PC, you should see a progress indicator to show how much of the file has been transferred.

When the transfer is finished, check that you received what you expected. If it what you expected, your HP<–>PC link works!

HP to HP Cable

If you want to use 9600 bps communication between two HP48s, then make two HP48 connectors and simply connect the two, swapping pins two and three.

```
HP to HP cable

HP#1 | HP#2
-----+-----
  1  |  1
  2  |  3
  3  |  2
  4  |  4
```

Warranty, Disclaimer, etc...

> Although the serial interface of the HP48 is protected internally, it is possible to damage the calculator when a wrong connection is made. I am not responsible for any errors in this file, or for any mistakes you may make.

From: Deborah Lynn Williams

I made an HP48 link out of four pieces of speaker wire and serial port plug. The wiring of the plug is available above. The connection to the HP plug is the difficult part.

I took 4 pieces of stranded speaker wire and cut them so that the wire and the insulation were even. I then took a paper clip and pushed it into the this open end, making a space between the wire and the insulation. I then had to trim some of the strands that were sticking out. I then just pushed this onto the pins in the HP48 port. It isn't a very strong connection, but it works fine if you don't jostle it.

The other ends of these speaker wires I connected to the serial plug. Just remember to label which wires go to which pin, or make them all different lengths.

From: John Cutter

Another cheap source for cables is the common serial mouse for PC-compatibles. I had a cheap one on which the button gave out after 2 weeks of use, so I opened it up. It had a serial 9 pin cable that disappeared into the mouse. Once open, there was a 4 pin 2 mm connector plugged into the mouse's circuit board. All I had to do was reverse two pins inside the connector, and it's been working fine since. I also noticed that Logitech mice have 6 pin 2 mm connectors, which could also be adapted. No soldering or crimping here!

12.2 Using a modem with the HP48

From: Diego Berge

My purpose here is to explain in some detail the steps you need to follow to be able to transmit data via modem with the HP48. Another related document is `<http://www.freeweb.org/freeweb/enrico/hp2modem.htm>` by Enrico Carta.

In order to successfully connect your HP48 calculator to a modem, the first thing you need is a 'Null Modem Cable'. A null modem cable is like a regular serial cable, except that two of its lines are crossed.

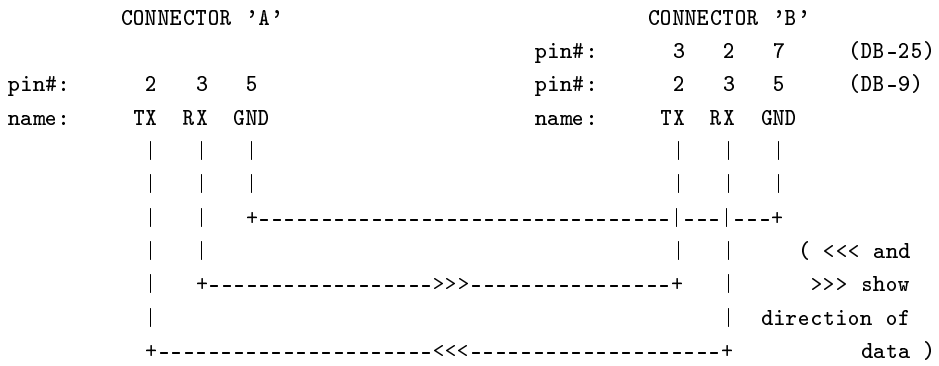
You might wonder: So why is it different connecting to a PC than connecting to a modem? Well, data communication devices can be classified in two groups: Devices which GENERATE or RECEIVE information, and devices which merely TRANSMIT information. The former are called Data Terminal Equipment (DTE), PCs are an example of DTE; the later is called Data Communication Equipment (DCE), and are primarily modems. Now, when PCs were designed, it seems like they didn't think somebody would ever want to hook two DTEs directly, as a result, in a typical PC all we have (besides the parallel printer port) is an RS-232 port, which is intended for connecting a DCE. So when a machine wants to connect to a PC, its port must be that of a DCE, even if it otherwise acts as a DTE. Within that class of hybrids fall HP48s, Psions, some (or most?) serial printers, mice, and a long etcetera.

Remember, the RS-232 communication scheme is:

```
DTE <--- DCE === DCE ---> DTE
```

If you want to link two DTE together, you must 'simulate' a DCE; that's what a Null Modem Cable is for. It's called 'Null' because it (the modem) does not actually exist, and 'Modem' because it acts like if there

was one. To build a Null Modem Cable for the 48, assuming that you already have a regular cable, you'll need two MALE RS-232C connectors, at least one of which must be a DB-9 (9-pin), and a short piece of mouse cable or similar (some 3 inches should do) (If you don't have a regular cable, I recommend that you get one). The wiring must be as shown here:



The pin number should be embossed near the pin itself in the connector.

Once the physical part is done, which in most cases will be the most critical, you're ready to try out your brand-new, state-of-the-art, $2.25 null modem cable. Plug it between your regular HP48 cable and your modem, turn the modem on (please don't forget this part :), and on the 48 type in:

```
[RightShift]+[""] [alpha] [alpha] AT [ENTER]
13 [LeftShift]+[CHARS] [CHR] [+]
```

In stack level 1 you should have (# is a little black square):

```
1: "AT#"
```

Now type:

```
[LeftShift]+[I/O] [NXT] [SERIAL] [XMIT] [BUFLEN] [DROP] [SRECV]
```

you should get:

```
2: "AT###OK##"
1: 1
```

If this test goes wrong, try two or three more times. If it still fails, check your cable(s), check that the 48 can transmit and receive to/from the PC, do the same with your modem. And when everything else has failed anyway, connect your modem to the PC, crank up a terminal program (as Terminal / Hyperterminal, or Kermit's 'Connect' command), and type:

```
AT&D0 [ENTER]
```

the modem should respond:

```
OK
```

This tells the modem to ignore the DTR signal from the host, which the 48 can't supply. Try again with the calculator, if it works, go back to the terminal and type:

```
AT&W0&Y0 [ENTER]
```

to save the current modem configuration as default. If it has not worked so far, I can't help you.

Assuming you've got it to work, now all you have to do is learn a few modem commands to dial and, possibily, hang up. In general, modem commands start with the two-character sequence 'AT' (no quotes) and end with a single <CR> character (dec 13), <LF> is optional.

To dial a number use:

```
ATDT12345<CR>
```

where 12345 should be replaced by the actual number. For example:

```
[RightShift]+[""] [alpha] [alpha] ATDT0800890011 [ENTER]
13 [LeftShift]+[CHARS] [CHR] [+] [LeftShift]+[I/O] [NXT] [SERIAL] [XMIT]
```

dials the AT&T Direct access number for the UK.

If your line is not connected to a digital switchboard, you may need to dial by pulses, then you should use instead:

```
ATDP12345<CR>
```

Next thing you'll probably ask is: 'How do I know that I'm connected?'

If the modem has successfully established a connection, it should respond in most cases with a message like:

```
CONNECT 9600/V34/LAPM/V42BIS/9600:TX/9600:RX
```

or simply "1" if its not in the default verbose mode. Also, if the modem has a 'CD' indicator, it'll usually light up. To find out via the calculator whether it's connected or a problem has occurred, use the following keystrokes:

```
( [LeftShift]+[I/O] [NXT] [SERIAL] )
[BUFLEN] [DROP] [SRECV]
```

repeatedly until a non-empty string appears in level 2. Some common messages are:

```
verbose:          non-verbose:

CONNECT                     1
NO CARRIER                  3
ERROR                       4
NO DIALTONE                 6
BUSY                        7
NO ANSWER                   8
DELAYED                    24
```

Once you get the CONNECT message, you're ready to send any data you want to the remote host as you'd usually do. I personally use Kermit this way when I'm not in the office or at home.

Finally, when you're done you'll want to hang up the line. The simplest way is turn the modem off, which I recommend. But if you want to instruct the modem to hang up, the process usually is:

(wait at least 1 sec without sending any data)

```
[RightShift]+[""] [alpha] [alpha] +++ [ENTER]
[LeftShift]+[I/O] [NXT] [SERIAL] [XMIT]
```

(wait at least 1 sec without sending any data)

```
[RightShift]+[""] [alpha] [alpha] ATH [ENTER]
13 [LeftShift]+[CHARS] [CHR] [+] [LeftShift]+[I/O] [NXT] [SERIAL] [XMIT]
```

Note, however, that this, as most of what I have said here about modem commands, might vary depending on each modem's particular brand, model and configuration. You may need to read your modem's manual for more details.

Disclaimer: As you'd expect, I take no responsibility for anything you may break, twist, cut, slice, burn, hurt, or otherwise damage while following these instructions.

12.3 Additional Information on the HP48 and RS-232

From: John Meyers

You can't connect the HP48 to anything else until you have first plugged in its cable, which finally brings the HP48's serial connections out to a DB9 connector. At that point, the HP48, including its attached cable, is clearly configured as DCE (the same as a modem), so you need a "crossover" (such as a "null-modem adapter") to connect the HP48 to a modem.

12.4 Using Non-HP RAM Cards

If you use RAM cards that are NOT designed for the HP48, it is possible to severely damage your HP48. If you want to be safe, you should only use RAM cards designed for the HP48.

Here is an edited discussion from comp.sys.handhelds.

```
From steveh@hpcvra.cv.hp.com Fri Mar  1 17:00:00 1991
From: steveh@hpcvra.cv.hp.com (Steve Harper)
Date: Thu, 10 May 1990 22:46:09 GMT
Subject: RE: HP48 SX Memory Card Pricing
Organization: Hewlett-Packard Co., Corvallis, OR, USA
```

There has been a substantial amount of comment regarding the memory cards for the HP48 SX and their prices. My purpose in this response is not to attempt to justify any particular price, but rather to present the technical reasons why there is a substantial price difference between the memory cards and other types of expansion memory for PC's, for example, with which users are probably more familiar.

Some users have correctly pointed out that the memory in the cards is static RAM rather than dynamic RAM commonly used in PC's. Dynamic RAM uses one transistor and a capacitor for each bit of memory whereas static RAM requires either four transistors and two resistors, or six transistors. The net result is that an equivalent amount of static RAM is much larger and therefore much more expensive than dynamic RAM. The advantage is that static RAM doesn't need to continually be running and drawing current (refresh cycles) to retain the contents of memory.

In addition, the static memory used in the cards is not just any static memory, but is specially processed and/or selected for very low standby current. This allows the backup battery in the card to keep memory alive for a very long time, rather than requiring the user to replace it every few months. The special processing and/or special testing to select low current parts adds to the already higher cost of the static RAM chips.

The standard molded plastic DIP package used for most integrated circuits, including memory chips, is relatively inexpensive because of its simplicity and the huge volumes. Unfortunately, these packages are too large to put into a memory card. Therefore, the card manufacturer mounts the individual silicon memory chips directly on a special thin PC board together with the memory support chips. Because multiple chips are being placed in a single hybrid package in a special process which has lower volume, yields are lower and this again causes the cost to be higher. Indeed, the yield becomes exponentially worse as the number of chips and interconnections increases in such a packaging process.

In addition to the memory chips themselves, two more integrated circuits and several discrete components are required for power and logic control. A bipolar technology chip senses the external voltage and switches the power to the chips from the internal keep-alive battery as needed. A CMOS gate array chip protects the memory address and data lines from glitches/ESD when the card is not plugged in. This chip also generates the proper enabling signals when there are multiple memory chips in the card, as is presently the case with the 128 Kbyte RAM card. These chips must be designed for extremely low current, just as the memory chips are.

In addition to the battery and the battery holder, the other mechanical parts are important, too. The molded plastic frame holds the PC board and provides the foundation for the metal overlays and the shutter-and-springs assembly which protects the contacts from ESD and from contaminants. The write-protect switch is also an important feature. It is quite expensive for the manufacturer to make the tools necessary to fabricate each of these parts as well as the tools to assemble and test the complete card. While the volume of memory cards is relatively low this tooling cost represents a significant part of the cost of each card.

Admittedly, there are other alternatives, such as those presently used in PC's, to provide a memory expansion capability. To provide that kind of expansion would require the calculator to be much larger than it is and possibly more expensive. This is clearly very undesirable.

Other features that were felt to be essential were the ability to distribute software applications and to share and archive/backup user-created programs and data. Other expansion alternatives do not provide these important benefits. The I/O capabilities of the calculator provide these features only to a limited degree.

One other item bears repeating here: Memory cards for use in the calculator will clearly indicate that they are for use with the HP48 SX. Other memory cards exist which are mechanically compatible with the HP48 S, but these cards cannot be relied upon to work electrically in the calculator. The HP48 SX cards are designed for a lower supply voltage range. Use of the other cards may cause memory loss, and under certain circumstances may even damage your calculator electrically.

```
From steveh@hpcvra.cv.hp.com Fri Mar  1 17:00:00 1991
From: steveh@hpcvra.cv.hp.com (Steve Harper)
```

```
Date: Fri, 11 May 1990 16:52:07 GMT
Subject: Re: Memory Card: Give Us True Facts!
Organization: Hewlett-Packard Co., Corvallis, OR, USA
```

My previous statement that under certain circumstances the calculator may even be damaged electrically is not a ploy. If the calculator's internal power supply voltage happens to be near the low end of the range, say 4.1 V, and the voltage at which the card's voltage control chip shuts it down happens to be near the high end of its range, say 4.2 V (this can and does occasionally occur for the non-HP48 SX cards), then the calculator will start to drive the memory address lines and the card will still have these clamped to ground (that's what it does to protect itself when there is not sufficient system voltage to run). This unfortunate situation may simply trash your memory, or if the calculator tries to drive enough of the lines high at the same time, several hundred milliamps may flow...for awhile that is, until something gives up... On the other hand, your calculator and a particular non-HP48 SX card may work just fine if those voltages happen to be at the other end of their ranges. These voltages are also slightly temperature sensitive. It may work in the classroom or office and not at the beach, or vice versa. The voltage trip point of the HP48 SX cards has been set lower (a different voltage control chip) so that this cannot occur, regardless of part and temperature variations.

One other item was brought to my attention yesterday by Preston Brown that I should have included in my original posting here. While most of us recognize that comparing RAM cards to a handful of dynamic RAM chips to plug into your PC is apples and oranges, it may be more interesting to compare the HP48 SX cards with cards for other products, like the Atari Portfolio, the Poquet, the NEC Ultralite, etc. I believe you will find that the prices on the HP48 SX cards are not at all out of line.

Steve "I claim all disclaimers..."

```
From prestonb@hpcvra.cv.hp.com Fri Mar  1 17:00:00 1991
From: prestonb@hpcvra.cv.hp.com (Preston Brown)
Date: Thu, 17 May 1990 17:26:53 GMT
Subject: Re: Memory Card: Give Us True Facts!
Organization: Hewlett-Packard Co., Corvallis, OR, USA
```

When the RAM cards detect that voltage is to low to operate they clamp the address lines to ground. This clamping is done by turning on the output drivers of a custom chip included on the card. The clamping current is specified to be 2mA min at the Vol output level. Since the 48 can be trying to drive the line all the way high even more current is typical. 10mA per fight is not uncommon with totals of several hundred mAs.

The VDD power supply is regulated at 4.1 - 4.9 with typical parts at the low end (4.3). The power to the cards is switched through a transistor, creating up to a 0.1V drop. Standard Epson cards have a significant chance of seeing this voltage as to low and shutting down. We have seen cards do this in the lab. When it occurs the calculator locks up with VDD pulled down to about 2.5V and 250mA being drawn from the batteries. This current drain greatly exceeds the ratings for the power supply and can damage your calc. The least that will happen is a loss of memory.

Now, why didn't we regulate VDD higher? The 48 has two power supplies VDD at 4.3 and VH at 8.5. VH cannot be regulated higher without exceeding the spec for our CMOS IC process. VH is used as the + voltage for the I/O. In order to meet a +3V output level VH must be more then 3.6V above VDD. (VDD is used as I/O ground). Our power supply system increase the battery life and reduces the cost greatly for the wired I/O.

Preston

12.5 Where can one obtain third party RAM cards?

If you decide you want to look into the non-mainstream alternatives for RAM cards, you might like to check the following URLs:

- <http://www.cynox.de/>
- <http://stolte-edv.com/de/>
- <http://www.cyberline.de/kkgbr/>
- <http://www.Digitalis.de/>
- <http://home.t-online.de/home/FFFFF/>
- <http://www.geocities.com/Eureka/Enterprises/3190/>

13 Appendix E: Where to get HP48 Programs

13.1 Best Programs and Where to Get Them List

This list is designed to help HP48 users locate useful, well written programs for their HP48, selected by HP48 users who send in recommendations for their favourite programs.

Compatibility is noted by either SX (SX only), GX (GX only), SX/GX (SX and GX), or SX?/GX (Works on GX, not sure about SX).

Most of the sites below require Internet access. However CompuServe has a modest HP48 archive in the HP handhelds forum. Type "go HPHAND" to access the forum. HP48 specific files can be found in DL 3. America Online also has an HP area you can access using "Keyword PDA".

Note that for most FTP sites, any programs in uploads directories are subject to movement into their appropriate directories (i.e. games or utilities), or removal by the archive maintainer. There is also the possibility that the program might have a slightly different name than the one given below, especially if newer versions are available. If information has changed that you know of, let me know so I can update the list.

I have also released my personal HP48 archive, which is updated occassionally. It should contain all of the following programs, as well as lots of others. More information can be obtained at: <http://www.engr.uvic.ca/aschoorl/archive/>

Applications

- **StringWriter** (SX/GX): This is a replacement editor by Jean-Yves Avenard. It offers such features as cut, copy, paste as well as a choice of fonts. Available at http://www.epita.fr/~avenar_j/
- **ChemLab** (SX/GX): A very fast, efficient, and complete periodic table from Arnold Moy. Available at <http://www.geog.ubc.ca/amoy/>

- **EXAL** (SX/GX): Great spreadsheet program by Al Arduengo. Available at <http://lonestar.texas.net/exal/>
- **GxTools** (GX): GxTools is in effect a "Toolbar" for your G/GX consisting of many useful utilities. Put together by Jack Levy and available at <http://www-personal.engin.umich.edu/levyj/hp48.html>
- **HP-Writer** (GX): This editor by Paquot Christophe is small, fast, and uses UFL FNT2 but still has all the important features. The current version is available at <http://www.geocities.com/SiliconValley/Lab/1930/>.
- **MiniWriter** (SX/GX): Small, fast, FNT1 text editor by Jean Yves Avenard. Available at http://www.epita.fr/~avenar_j/
- **Organizer** (SX/GX): Organizer is a library designed to act as a Personal Information Manager, or PIM. The goal is to create a high-speed, easy to use, compact, yet fully functional PIM for the HP48. Organizer includes a calendar, appointment scheduler, annual events, TODOs, and a phone manager. Also by Jack Levy and at <http://www-personal.engin.umich.edu/levyj/hp48.html>
- **Remote-BG** (SX/GX): Use your HP48 as a remote control! This is the best of the remote control programs and has the ability to learn new codes. By Bjorn Gahm, available at <ftp://nada.kth.se/home/d89-bga/hp/files/posted/rem34bg.zip>
- **Scribe** (SX/GX): Scribe is a library which will function as a Database or Rolodex. It is designed to hold a large amount of information in such a manner that it is easy to retrieve, quick to find, and transparent in usage. Another great program from Jack Levy at <http://www-personal.engin.umich.edu/levyj/hp48.html>

Documents

- **AUR Information** (GX): Information on the Advanced User Reference Manual compiled by Adrian Drury. Available at <http://www.contrib.andrew.cmu.edu/drury/hp48gaur/>
- **FAQ** (SX/GX): Contains valuable answers to Frequency Asked Questions. Read it! The latest version is available at <http://www.engr.uvic.ca/aschoorl/faq/>.
- **GX Programming Tips** (SX/GX): This document by Doug Cannon briefly describes how to write code that runs on both SX and GX. Available at <ftp://ftp.cis.com/pub/hp48g/programming/gxtips.zip>
- **Library Info** (SX/GX): Information on the structure and use of libraries. By Charlie Patton. Available on Goodies Disk 8.
- **ML Input** (SX/GX): How to get keyboard input from ML. By Joe Ervin. Available on Goodies Disk 7.
- **RPLMAN.DOC** (SX/GX): This document was written by HP and released without support to HP48 users. It describes how to program in System RPL. Available on Goodies Disk 4.
- **Saturn** (SX/GX): Saturn Processor Reference guide by Matthew Mastracci. The URL is <http://www.ucalgary.ca/mmastrac/saturn.doc>
- **SASM.DOC** (SX/GX): Also from HP, this is the unsupported guide to machine language programming for the Saturn processor. Available on Goodies Disk 4.

Games

- **Arkanoid** (SX/GX): Breakout clone with editor - by "HP Mad". SX version available on Goodies Disk 8. There is also a GX version floating around, try <http://www.hpcalc.org/games/arcade/arkanoid.zip>
- **Boulder DasHP** (SX): Classic game of Boulder Dash by David Presle "ORUS". Get the diamonds without getting smashed by falling rocks or running into enemies. Available on Goodies Disk 8.
- **Columns** (SX/GX): Tetris-like game by Spartacus. Available at <http://www.hpcalc.org/games/arcade/columns.zip>
- **Columns II** (GX): Columns with grey scale for GX only, again by Spartacus. Available at <http://www.hpcalc.org/games/arcade/columns2.zip>
- **Diamonds** (SX/GX): Great game by Doug Cannon! Control bouncing ball to get the diamonds. Available at <http://www.hpcalc.org/games/arcade/diam20a.zip>
- **Hapaman** (GX): PacMan clone by Davor Jadrijevic. Available at <http://www.hpcalc.org/games/arcade/hapaman.zip>
- **ICE Cube** (SX/GX): Great action arcade game in four levels of grey scale by Lilian Pigallio. Available at <http://www.hpcalc.org/games/arcade/ice1_1.zip>
- **Jytris** (SX/GX): One of the best tetris games including music, cartoons, bombs, and grey scale intro. Written by Laurent Jouanneau. Separate libraries are available for all HP48s (including minimized libraries for S and G). Available at <http://www.hpcalc.org/games/arcade/jytris11.zip>
- **La Mine aux Diamants** (SX/GX): Another Boulder Dash clone. By Lilian Pigallio. Available at <http://www.hpcalc.org/games/arcade/diamants.zip>
- **Lemmings** (SX/GX): Save the lemmings from killing themselves! By C. de Dinechin. Available at <http://www.hpcalc.org/games/arcade/lemmings.zip>
- **LodRunner** (GX): LodeRunner clone for the HP! By HPOWER, with improvements from Mika Heiskanen. Available at ftp.cis.com.
- **Phoenix** (SX/GX): Phoenix clone by Davor Jadrijevic. Available at <http://www.hpcalc.org/games/arcade/phoenixg.zip>
- **Quick Reversi** (SX): Fastest Reversi by Leon Pavlov. Quite smart! Available on Goodies Disk 8.
- **Tetris** (SX/GX): Tetris clone - small, fast, great! By Detlef Mueller. Available at wuarchive.wustl.edu.
- **SCHIP** (SX/GX): Video game interpreter for SCHIP designed games. By Erik Bryntse. SX version available on Goodies Disk 3. Available at <http://www.hpcalc.org/games/schip/>
- **Vaders** (SX/GX): Space Invaders by Joe Ervin. Available at <http://www.hpcalc.org/games/arcade/vaders21.zip>
- **WarioHP Land** (SX/GX): Adventure game similar to Mario Land. By Julien Meyer and available at <http://perso.club-internet.fr/meyert/wariohp/wariohp.zip>
- **Willy** (SX/GX): Side scroller game by Mario Mikocevic. Available at <http://www.hpcalc.org/games/arcade/willy.zip>

Graphics

- **4 Colors** (SX/GX): Utilities for making four shade greyscale graphics. By Lilian Pigallio and available at <http://www.hpcalc.org/graphics/grayscale/4colors.zip>
- **Claudia** (SX/GX): Greyscale demo of Claudia Schiffer, the supermodel. By Mohamed Fatri. Available at <http://www.hpcalc.org/graphics/grayscale/schiffer.zip>

Math

- **ALG48** (SX/GX): Very fast algebraic manipulation utilities by Mika Heiskanen and Claude-Nicolas Fiechter. Available at <http://www.cs.pitt.edu/fiechter/hp48/>
- **CNTR** (SX/GX): Discrete and continuous control systems design; Inverse Z and Laplace Transforms. By Cesar Crusius. Available at <http://www.hpcalc.org/math/symbolic/cntr300.zip>
- **Erable** (SX/GX): Tons of powerful math utilities! By Bernard Parisse. This package is a successor to the ALGB package by the same author. The latest version is at <ftp://fourier.ujf-grenoble.fr/pub/hp48/erable.zip>
- **Math** (SX/GX): Symbolic/numerical matrices and polynomials by Cesar Crusius. Available at <http://www.hpcalc.org/math/symbolic/math210.zip>
- **Matrix** (SX/GX): Very fast matrix writer replacement by Arnold Moy. Handles real and complex matrices, and uses the UFL small font for its display. Available at <http://www.geog.ubc.ca/amoy/>
- **MTools** (SX/GX): MathTools by Jack Levy. Faster and friendly replacements for the HP's math functions. Available <http://www-personal.engin.umich.edu/levyj/hp48.html>
- **QPI** (SX/GX): Replaces ->Q, shows quotients, and functions of pi, radicals, LN and EXP. Very useful! The latest version can be found at <http://www.engr.uvic.ca/aschoorl/>
- **SolveSys** (GX): Solves systems of equations. By Sune Bredahl. Available at <http://www.hpcalc.org/math/symbolic/ssys402.zip>
- **STAT03** (GX): STAT03 is a collection of fairly fast statistical/matrix utilities. It contains 49 more or less useful commands. About 25% is written in ML. The author is Christian Meland, and it is available at <http://www.hpcalc.org/math/statistic/stat03.zip>

PC

- **HP-IDE** (Win32): HP-IDE is a SysRPL/ML IDE for Win32, making directory and library creating much easier to do (completely on your PC). Available at <http://www.acs.ucalgary.ca/mmastrac/hp-ide/>
- **HP48 Explorer** (Win32): This program runs in Windows 95 and helps simplify file transfers. It is available at <http://members.tripod.com/nberry/hp48g/index.html>
- **HPShell** (DOS): The HPShell is a development environment for HP48 programming on PC. It handles transfers as well as editing and character translation. Available at <http://www.kst.dit.ie/people/twellige/hpshell.html>
- **ROMDump Wizard** (Win32): The best ROM dump program available. By Christoph Giesselink, and available at <http://privat.swol.de/ChristophGiesselink/>

Programming

- **Jazz** (SX/GX): This is best System RPL and Machine Language development library that runs right on your HP48! Written by Mika Heiskanen. The Jazz library provides commands for assembling, disassembling and debugging both System RPL and machine language. Both full and light versions are available at <http://www.engr.uvic.ca/aschoorl/>
- **HP's Tools** (DOS): Development tools for System RPL and libraries on the PC. On Goodies Disk 4.
- **GNU-Tools** (Unix, DOS): A set of replacements for HP's development tools by Mario Mikocevic (Mozgy) which that not only runs on DOS, but also typical Unix systems such as Linux and Solaris. It is distributed in both source and binary form.

  Many of the larger HP48 programs like Jazz, Java, and the UFL were developed with GNU-Tools. It is available at:

  - <ftp://gnjilux.cc.fer.hr/pub/hp48/gtools/>
  - <http://www.zems.fer.hr/mozgy/jwz/hp48.html>
  - <http://www.engr.uvic.ca/aschoorl/>
- **SSS** (SX/GX): The SysRPL Shell by Stefan Wolfrum provides an interface that helps you program in System RPL. Available at <http://www.hpcalc.org/programming/misc/sss22.zip>

Utilities

- **BKUP** (SX/GX): BKUP is written entirely in System RPL and is intended to make a backup of any port(s) the user specifies. By Bob Apodaca, available at <http://www.hpcalc.org/utils/memory/bkup46.zip>
- **FF** (GX): This File Finder is designed for users with large directory structures. It accepts a string and will search your directories recursively for matches. By Jorge Costa and available at <http://alfa.ist.utl.pt/l38861/>
- **FIXIT** (SX/GX): This program by Mika Heiskanen and Joe Horn repairs "HPHP48-" bad downloads. Available in ASC format in the FAQ as well as on Goodies Disk 8.
- **FXRECV** (GX): Fixes bug in X-Modem for G/GX. Not necessary for Rom R. By Cary McCallister. Available on Goodies Disk 9.
- **HP82240B Emulator**: Allows you to emulate the HP82240B IR Printer. Written by Jarno Peschier and at <http://www.students.cs.ruu.nl/people/jpeschie/>
- **PCT** (SX/GX): Both SX and GX versions are available of this file and library maintenance program. Available at the standard sites. Be sure to use the correct version for your machine! In particular the TREE.PCT file is an S/SX program that is a common cause of crashes on G/GX machines.
- **UFL** (SX/GX): The Universal Font Library provides a centralized location for common fonts between programs. Also comes with a Font Maintenance Library (FMnt). Maintained by Andre Schoorl and available at <http://www.engr.uvic.ca/aschoorl/ufl/>
- **USEND/URECV** (SX/GX): Significantly faster IR file transfers! Approximately 4 to 6 times faster than SEND. By Dan Kirkland. Available at <ftp://ftp.hp.com/pub/calculators/hp48g/comms/usnd3.dk>
- **STOFIX** (GX): Fixes STO bug on Roms L and M. By Detlef Mueller. Available at <http://www.hpcalc.org/utils/memory/stofix.zip>

Sound

- **Tricorder** (SX/GX): This program will make your HP48 beep when it is held close to an object using the IR port. Great for amusing friends! Written by Steven Ourada, improved by Richard Steventon, and available at <http://www.hpcalc.org/utils/comms/wtrico.zip>

Stack Replacements and Related Programs

- **EQStk** (SX/GX): 7-line replacement stack intended for mathematics. View objects on the stack in their native form. By Mika Heiskanen. Available at <http://www.engr.uvic.ca/aschoorl/>
- **Java** (SX/GX): The totally improved 5-line stack environment! Includes lowercase menus, improved status area, on-stack algebraics, improved interactive stack, configurable font, scroller, fast ML routines and more. By Richard Steventon, Andre Schoorl (hey that's me!), and Will Laughlin. The latest released version (3.2a) is available at <http://www.engr.uvic.ca/aschoorl/>
- **MetaKernel** (GX): This is a total replacement for a lot of built in HP48 ROM code, including the stack, editor, equation writer, matrix writer, and picture environments. It also includes SysRPL and ML development tools. A lot of work has been put into it to make everything extremely fast. The MetaKernel requires a GX and is sold on a ROM card. Documentation is available in both French and English. The full version must be purchased from the authors, but a demo version is available at <http://www-miaif.ibp.fr/gerald/mk.html>

13.2 Other Web / FTP Sites

- <http://www.hpcalc.org/>
- <ftp://ftp.hp.com/pub/calculators/>
- <ftp://ftp.stud.fh-heilbronn.de/>
- <ftp://ftp.cyberbox.de/pub/hp48/>
- <ftp://ftp.cyberbox.de/pub/hp48/fileecho/>
- <ftp://hplyot.obspm.fr/hp48/>
- <ftp://wuarchive.wustl.edu/systems/hp/hp48/>
- <ftp://ftp.ee.ualberta.ca/pub/HP48/>
- <ftp://oak.oakland.edu/pub/ham/hp48/>
- <ftp://ftp.polymer.uakron.edu/pub/hp48g/>
- <ftp://nic.funet.fi/pub/misc/hp48g/>
- <ftp://nada.kth.se/home/d89-bga/hp/>
- <ftp://ftp.fht-mannheim.de/pub/hp48/>

13.3 FTP by Electronic Mail

For those of you who don't have FTP access, you can FTP programs by mail. To do this, you send commands via e-mail message to the ftpmail server, and it then sends back the programs, usually in a uuencoded format by e-mail.

For access to any ftp site, I recommend the sunsite ftpmail server. To get more information, send a mail message to <*ftpmail@sunsite.unc.edu*> with the contents of:

```
help
quit
```

There are other FTP by mail servers like the sunsite server, but they are more heavily used, and take longer.

13.4 Bulletin Boards with HP48 Programs

Official Hewlett Packard BBS (48)

- Phone: (208) 344-1691 ; Corvallis, Oregon.
- Second Node: (541) 715-3277 ; Speed: 9600 baud

Foundations in Science (48/100)

- Phone: (206) 562-7083 644-2723 ; Speed: 14.4
- Contact: <*civan@eskimo.com*>

UAH ACCESS BBS (28/48/95/100)

Phone: (205) 895-6152 ; U. Alabama Huntsville.

The Ninth Bit BBS (48)

- Phone: (405) 372-7204 ; Speed: 14.4
- Contact: <*scott.wilkins@ninth.ucc.okstate.edu*>

Raiders of Lost Arg (48/95)

- Phone: (416) 968-6633 ; Speed: 38400 N81
- Contact: <*john.ricciuti@can.rem.com*>

NES BBX (48)

Phone: (503) 640-4263 ; Goodies Disks in File Area 22

[(hp)]al BBS (48/95/100)

- Phone: (612) 642-5906 ; Features: Usenet, E-Mail
- Contact: <*rab@lydia.mn.org*>

Electronic Design News BBS (48)

Phone: (617) 558-4582 ; Features: Open to All

Nybble's Byte (28/48/95/100/etc)

- Phone: (708) 304-0666 ; Speed: 28.8
- Features: 24 hr. Usenet/Internet

Terrapin Station (48)

Phone: (708) 657-9543

48 Files BBS (48)

- Phone: (801) 371-8760 ; Speed: 14.4
- Features: Over 3000 files

The ONE BBS (48)

Phone: +33-149-887-691 ; France.

Cyberbox (48)

- This German BBS has a filebase that contains about 1000 Files, 30MB. All are zipped, with descriptions.
- Contact: Patrick.Werner@cyberbox.de
- Phone:
  - Line 01-04 : +49-441-3990033 USR 2400-33600
  - Line 05-06 : +49-441-9396977 ISDN X.75
- There is also full access to the filebase at `<http://www.cyberbox.de/software/hp48/>`. The start page for all file areas is `<http://www.cyberbox.de/files/index.html>`

13.5 HP Goodies Disks

The HP Goodies Disks are a set of disks that are chock full of great HP48 files. They are collected by Joe Horn, and are available on the Internet. They contain all shareware/freeware files. Currently there are 11 disks. They can be obtained via anonymous ftp from the places listed below.

In the listings above, they are referred to by "GD #". All the goodies disks have a file "fileinfo.src" which contains an index of all the files. I would recommend everybody looking through all these disks, as there is lots of great stuff on them!

- `<http://www.hpcalc.org/compilations/horn/>`
- `<ftp://ftp.ee.ualberta.ca/pub/HP48/>`
- UAH BBS: (205) 895-6152
- CompuServe: go HPHAND ; DL 3

14 Contributors

The maintainer thanks the following international contributors (covering a total of 16 countries) for their invaluable help in compiling this FAQ:

U.S. and Canada:

Dave Arnett, Allen Arnold, Diego Berge, Rich Brown, Doug Cannon, Andrew Chen, Dallan Christensen, James Cook, John Cutter, Andrew Dooley, Chuck Dinsmore, Jeff Dubin, Egan Ford, John Goerzen, Paul Hart, Joe Horn, David Johnson, Jack Levy, Keith Maddock, Scott Marlowe, Thomas De Matteis, John Meyers, David Moisan, Derrik Pates, David Peterson, Othniel Rawlins, Eric Rechlin, David Rice, Ray Richardson, Maynard Riley, Jake Schwartz, Jeremy Smith, Chris Spell, Lee Studley, Deborah Lynn Williams, Mark Wilson, Patrick Yagle.

Jean-Francois Larin, Arnold Moy, Jeff Sketchley, Matt Willis, Ryan Youck.

Europe:

Rachid Benzaoui (France), Jorge Costa (Portugal), Yves Gadioux (France), Bjorn Gahm (Sweden), Maarten van den Hoek (Netherlands), Pieter-Bas IJdens (Netherlands), Peter Karp (Germany), Jens Kerle (Germany), Wlodek A C Mier-Jedrzejowicz (U.K.), Mario Mikocevic "Mozgy" (Croatia), Makoto Miyamoto (Germany), Bernard Parisse (France), Matthias Paul (Germany), Jarno Peschier (Netherlands), Lilian Pigallio (France), Rob Simpson (U.K.), Klaus Wolferts (Germany), Stefan Wolfrum (Germany), Stevens Wouter (Belguim), Matjaz Vencelj (Slovenia).

Elsewhere:

Guido Carvajal (Chile), Cesar Crusius (Brazil), Kevin Cheng (Australia), Antonio Pacheco (Venezuela), Alejandro Arrieta Rios (Chile), Richard Steventon (South Africa).