

HP Forum Archive 13

[[Return to Index](#) | [Top of Index](#)]

RPN/RPL - What?

Message #1 Posted by [Jack](#) on 9 June 2003, 11:44 p.m.

Did HP swich from RPN to RPL at some point or was RPL used only on certain models. Is RPL an official HP term?

My HP 12C manual(The only manual I can find with out unpacking a garage full of boxes) doesnt refer to RPL, but then it doesnt mention RPN either. However, the HP 100LX still refered to setting the calculator to "RPN Mode"

Did I miss a turn somewhere?

Jack

Re: RPN/RPL - What?

Message #2 Posted by [Ernie Malaga](#) on 10 June 2003, 2:19 a.m.,
in response to message #1 by Jack

Quote:

Did I miss a turn somewhere?

Yep. Around 1987 HP announced the HP-28C, which uses RPL. I am not sure whether the financial HP-18C uses it too (I've never seen one), and the 18C is, I believe, a bit older than the 28C.

In any case, RPL is a bona fide HP acronym, and it names an operating system similar to, but different from, the classical RPN. Other calculators that use RPL are the 48 and 49.

What makes RPL different from its predecessor is (1) the stack is infinite in size. Gone are X, Y, Z, T; now stack levels are referred to by number. (2) Stack registers can contain any kind of data -- even programs. (3) [ital]Everything[/ital] is backwards: not only 2 ENTER 3 +, but also "ABC" STO, which stores Level 1 into variable "ABC".

There is no consensus among users whether we like RPL better than RPN. I'm in the "no" side, but I seem to be in a minority.

-Ernie

Re: RPN/RPL - What?

Message #3 Posted by [Valentin Albillo](#) on 10 June 2003, 7:27 a.m.,
in response to message #2 by Ernie Malaga

Ernesto wrote:

There is no consensus among users whether we like RPL better than RPN. I'm in the "no" side, but I seem to be in a minority.

Count me in.

Best regards.

Re: RPN/RPL - What?

Message #4 Posted by [Massimo Gnerucci \(Italy\)](#) on 10 June 2003, 8:27 a.m.,
in response to message #2 by Ernie Malaga

Quote:

There is no consensus among users whether we like RPL better than RPN. I'm in the "no" side, but I seem to be in a minority

me too... ;-)

Massimo

Re: RPN/RPL - What?

Message #5 Posted by [R Lion](#) on 10 June 2003, 8:33 a.m.,
in response to message #2 by Ernie Malaga

Quote:

There is no consensus among users whether we like RPL better than RPN. I'm in the "no" side

Hi from the "yes side" ;-)

I'm a fanatic, lover, maniac, etc.,etc. of the 48... BUT I wouldn't say that "like" is my feeling about RPL: I simply find easily the way for my programs do what I aimed for.

I'm not a programmer, so I'm always sure that my programs could be optimized for any more expert, but they do the work.

This calculator do all I have requested "her".

That's what I need.

Raul

I second Raúl..

*Message #6 Posted by **Vieira, Luiz C. (Brazil)** on 10 June 2003, 10:07 a.m.,
in response to message #5 by R Lion*

Hi, Jack! (Sounds familiar... sorry, I could not resist :)

I second Raúl and I'm at the "YES" side, but I'd never disagree with others choices. I like structured programming and functional paradigm. I would never give-up RPN, and RPL implemented functional, structured programming to an enhanced level.

I'm not calling RPL as an enhanced RPN. Not that I do not think that way, it's to avoid flames and blames.

Most of all, **personal** programming is a matter of choice and taste. I'm completely and 100% sure any RPN programmer is able to understand and use RPL as well; the reasons many in here decided not to go for it are their reasons. It's not the same case, but may be "seen" as a personal choice: RPL-only programmers, those who did not get RPN at the time it was a fever, may also not like (or even feel tied for restrictions) to use RPN. Sometimes, when programming in an RPN calculator, I miss another stack register, or a PICK command, or DUP2, or an extra ALPHA register... well, they're not there, I must find another way.

My 2¢.

Luiz C. Vieira - Brazil

Re: RPN/RPL - What?

*Message #7 Posted by **jimc** on 10 June 2003, 11:23 a.m.,
in response to message #5 by R Lion*

I think I'd like RPL a little more if only there was some reasonable documentation! My 48's information on how to program lies somewhere between "inane" and "you really don't want to do this do you?". I am not a programmer by nature or by occupation. I just want an answer. As it stands, I am in the RPN camp.

BTW, if anyone could point me to a GOOD reference to learn RPL from (ideally from the 'net), I'd really appreciate it! It might even entice me into not listing my 48 on Ebay!;

Thanks.

Learning RPL

*Message #8 Posted by **R Lion** on 10 June 2003, 11:46 a.m.,
in response to message #7 by jimc*

Quote:

BTW, if anyone could point me to a GOOD reference to learn RPL from

Well: the Advanced User's Reference Manual (AUR) is THE book (IMHO)

Raul

Re: RPN/RPL - What?

*Message #9 Posted by **bill platt** on 10 June 2003, 12:12 p.m.,
in response to message #7 by jimc*

Jim,

I found some very good references at www.hpcalc.org

I have three 5 references copied down to my hard drive at home; I would be happy to e-mail them to you, or put them on my FTP site later.

Three are on my hard drive here at work:

- 1.) The 48gx users manual (big, PDF)
- 2.) "An Introduction to Programming the HP 48G/GX Calculator" by Mervin E. Newton (PDF file, 135 kB)
- 3.) "Programming In User RPL" by Eduardo M. Kalinowski (225 kB, PDF)

Re: RPN/RPL - What?

Message #10 Posted by [Valentin Albillo](#) on 10 June 2003, 1:12 p.m.,
in response to message #7 by jimc

jimc posted:

"I am not a programmer by nature or by occupation. I just want an answer."

That's exactly my point. Classic HP calculators programmable in RPN were designed as tools for technical professionals, like engineers for example, who weren't programmers nor did they want to be, they just needed *answers*. Simple keystroke RPN programming provided those answers *easily and fast*.

On the other hand, RPL is a full-fledged, complicated, object-oriented language, which requires the user to deal with the vagaries and complications necessarily associated with such an OOP language, thus negating all simplicity and requiring its users to become what they'd rather not.

The result ? A divided user base, some loving RPL, some hating it, to the benefit of noone. Most technical users would find RPL unsuitable to get solutions fast, and would look elsewhere, ultimately dismissing RPL for good. The RPL revolution would gain only a very small percentage of people anew, while losing a lot of old timers. Most people saw the HP48/49 machines as *"already too complicated"* and they indeed are. I'm still waiting for someone to claim that any given RPN machine is *"too complicated"*. Even the HP-41CX, with its large instruction set, its many ROM modules, and even its synthetics, still remained perfectly 'understandable', easy to grasp and use. Try that with an RPL machine.

I really feel classic RPN realized the best compromise between simplicity and power, with its 4-level+LastX RPN stack. Three or two level would be insufficient. More than 4 levels would be a book case of diminishing returns. Throwing it all in exchange for infinite levels which could also unexpectedly become *three, two, one, or none (!)*, with no automatic replication, with an overblown "last x" which could also be *non-available*, with all store operations now forced to always being *indirect*, etc, etc, was too high a price to pay for the loss of simplicity and convenience.

In the end, it caused most customers to completely ignore (and dislike) RPL, and RPN suffered as well. Every paradise has its snake.

"BTW, if anyone could point me to a GOOD reference to learn RPL from (ideally from the 'net), I'd really appreciate it!"

IMHO, this is the best, easiest, clearest book on the subject for learning purposes, assuming an user proficient with RPN but not versed at all in RPL:

"HP 48 Programming Examples" (HP Press Series) by Donald R. Mackenroth, paperback, 341 pages, Addison-Wesley December 1991, ASIN: 0201563258)

It includes lots and lots (more than 100) sample programs of all kinds, thoroughly explained and commented by the author, who, BTW, also wrote the excellent HP-25 Owner's Handbook, among others.

Best regards.

Sure I understand your point...

Message #11 Posted by [R Lion](#) on 10 June 2003, 2:01 p.m.,
in response to message #10 by Valentin Albillo

...because I often think that is "impossible" to change my 48GX for the 49G or whatever new calculator or system or expander... When someone feels comfortable, is very difficult taht he wants to change.

Raul

Re: RPN/RPL - What?

Message #12 Posted by [Ernie Malaga](#) on 11 June 2003, 12:17 a.m.,
in response to message #10 by Valentin Albillo

Valentin:

Quote:

That's exactly my point. Classic HP calculators programmable in RPN were designed as tools for technical professionals, like engineers for example, who weren't programmers nor did they want to be, they just needed answers. Simple keystroke RPN programming provided those answers easily and fast.

On the other hand, RPL is a full-fledged, complicated, object-oriented language, which requires the user to deal with the vagaries and complications necessarily associated with such an OOP language, thus negating all simplicity and requiring its users to become what they'd rather not.

I agree with you wholeheartedly. When I was in college a friend had an HP-65. I taught him how to program it and, although he had absolutely no prior experience with computers, he understood everything the minute I said to him, "A program is nothing more than a series of codes that tell the calculator which keys to push -- automatically from the inside. Just imagine the calculator is pushing its own keys."

Boy, was that a revelation! Soon he was writing programs like a pro. [b]That's[/b] what RPN did for users -- provide a simple and efficient environment for either manual or programmed operation.

No longer true with RPL. Now the user is expected to learn the use of local variables, embedded programs, and an incredibly extensive list of operations, functions and commands -- for which there is little documentation.

Many die-hard RPL users scorn RPN calculators because they have so little RAM. Well, part of it is because RPN machines don't use up so much room in memory. Most instructions take up a single byte!

I'm not saying that RPN is perfect. I'm not even attempting to imply it. It has several serious defects (being GTO-happy is the most egregious), but for simplicity and ease of learning, it can't be beat.

-Ernie (PPC# 6594)

Re: RPN/RPL - What?

Message #13 Posted by **Vieira, Luiz C. (Brazil)** on 11 June 2003, 1:35 a.m.,
in response to message #12 by Ernie Malaga

Hi, Ernie;

Yours are thoughtful words, but do you allow me to add a few comments to yours?

Quote:

"A program is nothing more than a series of codes that tell the calculator which keys to push -- automatically from the inside. Just imagine the calculator is pushing its own keys." (...) Now the user is expected to learn the use of local variables, embedded programs, and an incredibly extensive list of operations, functions and commands -- for which there is little documentation.

This is true when you're talking about RPN and AOS 80's pocket calculators. At the time BASIC and a few C-based pocket appeared, this rule no longer applied to them, too. As I mentioned sometime ago, I take RPN as a user interface, and programs using RPN interface take the advantage of the interface. The first RPN calculators were not even programmable, and they were RPN calculators.

About the little documentation I have nothing to say because I don't know all of them; anyway I have at least seven non-HP RPL-related publications, not counting the one I wrote about RPN-RPL "transition" (I must be careful not using Hicks' title...). But let's face one important fact: RPL stands for Reverse Polish *Language*. A language must be learned in its structures, as any other language. And in most cases (almost all) there is no way to use some languages' specific resources directly through the keyboard. There are exceptions, of course, but programming languages are intended to be used in programs. RPN is not a language; as the name says, it's a notation. And keystroke sequences in RPN-based calculators work fine as programs till the moment you test flags, use labels and gotos, subroutine calls and returns, index registers and control loops... These are of no use when performing direct computations, and you have to abstract when learning them as well as in any other language.

Using an RPL-based calculator as an RPN calculator are the first steps taken in their own user's manuals to introduce them:

number [ENTER] number [operation]

Or in a general, RPL-related form:

object [ENTER] object [command]

Using their programming structures and resources demand reading.

Quote:

Many die-hard RPL users scorn RPN calculators because they have so little RAM. Well, part of it is because RPN machines don't use up so much room in memory. Most instructions take up a single byte!

In any processor architecture, the combination of n bytes for instructions lead to an enhanced set of instructions. RISC and CISC architectures are the sign this is a subject for further studies. PIC processors have a 12-bit deep instruction code, allowing 16 times more codes than the possible 256 with 8-bits only. If you take an HP11C and write down all possible, valid combinations of all keys and their respective prefixes, there is no way to go beyond 256. That's the maximum achieved. In other hand, if you do the same with the HP15C, you'll find there are more than 256 possible combinations. Then it's time to combine codes and select which one will be the prefix code, the one to be combined with the others, the code that will set a second-bank ROM call. Those who programmed Z-80 may remember the prefix codes: I only remember CEh, if I am not wrong.

The use of a single byte in program lines surely enhances memory use, but the fact is that memory was too expensive when programmable calculators first appeared. And some calculators, like the HP55, HP65 and both TI58/59, could merge a few combinations of codes, causing memory space to be consumed fast. And this is another reason I do not like to associate RPN (or even AOS) to programming resources. When we are talking about program-related structures, either RPN or RPL have their own environments, and both need a different approach. You may find both RPL- and RPN-calculator's users that do not know how to program. And both may use their calculators as well.

I know these are considerations based on some tendencies of mine, no doubts about. I just want to express what I see about RPN and RPL, mostly that RPL and RPN are related to each other, and that I repeated some thoughts in a different fashion, and for this I'm sorry. I think I must stop now to wait for comments, blames, flames, suggestions...

Luiz C. Vieira - Brazil

Edited: 11 June 2003, 2:38 a.m.

Re: RPN/RPL - What?

Message #14 Posted by [Valentin Albillo](#) on 11 June 2003, 7:24 a.m.,
in response to message #13 by Vieira, Luiz C. (Brazil)

Hi Luiz & al.:

Luiz posted:

"I take RPN as a user interface, and programs using RPN interface take the advantage of the interface. The first RPN calculators where not even programmable, and they were RPN calculators."

I think you're confusing terminology. There's no way RPN can be called "a user interface". That would be a keyboard or wand on the hardware level, or a windowing system with icons and menus on the software level. Else, you could call BASIC a "user interface" to the underlying algebraic system. Perhaps you should refer to the RPN programming paradigm by some other name if you're going to consider it a language (I think HP-41Cs brand of RPN programming was called "FOCAL" at some point).

"In other hand, if you do the same with the HP15C, you'll find there are more than 256 possible combinations."

Many, many more. Just considering STO type instructions, there are more than 150 possible. Similarly for RCL instructions.

And some calculators, like the HP55, HP65 and both TI58/59, could merge a few combinations of codes, causing memory space to be consumed fast.

You meant "slow" instead of "fast", right ? Actually, merging codes *saved* memory space, of course; memory space would be consumed fast when keycodes *weren't* merged, as in the HP-55.

"You may find both RPL- and RPN-calculator's users that do not know how to program."

Anything goes in this bizarre world we've got, but I think that using an actual, nowadays existing, RPL calculator *only* for casual 'hand' computations, i.e., no programming at all (because its user "does not know how to program"), is really, really, *overkill*. For such use, I'd rather stick to my HP-11C or 15C, thank you very much. They just don't *consternate* me. :-)

Best regards.

Re: RPN/RPL - What?

Message #15 Posted by [Vieira, Luiz C. \(Brazil\)](#) on 11 June 2003, 7:59 a.m.,
in response to message #14 by Valentin Albillo

Hi, Valentin;

maybe I'm confusing something, I don't know. I call interface whatever exists between (inter) two systems, allowing them to "communicate" with each other. I consider both SW and HW resources necessary to compose an interface. If you take the automobile's cockpit as an interface, it doesn't mean anyone can drive a car, because there is a necessary protocol (driving skills) to make it "run". The cockpit MMI is complete only when there is a human being using it to drive the car, "communicating" with it. Likely, there are also machine-to-machine interfaces, being Hewlett-Packard's Interface Loop - HPIL - one of the very well known. There are both hardware and protocol needs to any equipment that wants to connect to another by means of an HPIL. If I am wrong it's because I'm taking the term in a much general meaning.

Best regards.

Luiz C. Vieira - Brazil

Re: RPN/RPL - What?

Message #16 Posted by [Valentin Albillo](#) on 11 June 2003, 9:37 a.m.,
in response to message #15 by Vieira, Luiz C. (Brazil)

Luiz posted:

"Maybe I'm confusing something, I don't know. I call interface whatever exists between (inter) two systems, allowing them to "communicate" with each other."

Agreed. My objection was that you didn't use just "interface" on its own, which is as you just defined, but "user interface", which is rather a different beast in computer jargon.

Else, everything would be a "user interface", including RPN, your love for your wife, etc, etc.

Anyway, very interesting discussion, thanks for it.

Best regards.

User interface × User-level interface

Message #17 Posted by [Vieira, Luiz C. \(Brazil\)](#) on 11 June 2003, 10:57 a.m.,
in response to message #16 by Valentin Albillo

Hi, Valentin;

not to extend the discussion more than enough, after posting it and got the road (I'm back and alive...) I thought I should post "user-level interface" instead of "user interface". At the moment it occurred to me user interface is something created by the user, than you're right complaining: RPN is not a user interface.

I'd rather writing user-level interface, what would better explain how do I see RPN.

Best regards and thank you too, Valentin; it's indeed a "teasing" subject.

Luiz C. Vieira - Brazil

Edited: 11 June 2003, 12:17 p.m.

Re: RPN/RPL - What?

Message #18 Posted by [Wayne Brown](#) on 11 June 2003, 9:08 a.m.,
in response to message #14 by Valentin Albillo

Quote:

Anything goes in this bizarre world we've got, but I think that using an actual, nowadays existing, RPL calculator only for casual 'hand' computations, i.e., no programming at all (because its user "does not know how to program"), is really, really, overkill. For such use, I'd rather stick to my HP-11C or 15C, thank you very much. They just don't consternate me. :-)

I use a lot of RPL programs *other* people have written (like MetaKernel and Erable) on my 48GX, but I very seldom write any programs on it myself (mostly because I haven't yet been interested enough to devote the time necessary to learn more than a small subset of RPL programming). However, I frequently throw together small, special-purpose (often one-time-use) RPN programs on my 16C and 41CX. I just prefer the keystroke programming of the older models over RPL, though I really like the other features of my 48GX.

My day-to-day work involves dealing with high-level programming languages and I get enough of that at the office, so I've been putting off delving into the depths of RPL, which so far I haven't enjoyed very much. For anything except extremely trivial programs I prefer to go to hpcalc.org and grab whatever package is closest to what I need.

Re: RPN/RPL - What?

Message #19 Posted by [Massimo Gnerucci \(Italy\)](#) on 11 June 2003, 10:37 a.m.,
in response to message #18 by Wayne Brown

Quote:

My day-to-day work involves dealing with high-level programming languages and I get enough of that at the office, so I've been putting off delving into the depths of RPL...

Right on the nail!

Who needs *another* computer language?

IMHO a sort of macro language is all what is needed in a calculator for trivial, everyday, use.

When more specialized needs arise you either have to spend a great deal of time delving in the innards of UserRPL/SysRPL/ML/SynthProg and the like, or searching online for a library/listing that fits your needs or resort to a more specialized device.

It may be fun studying another language... when you have time and you can practice it everyday. No fun at all when time is **not** on your side and have to remember another way to write down an if/then/else.

My 1€¢,
Massimo

Re: RPN/RPL - What?

Message #20 Posted by [Ellis Easley](#) on 18 June 2003, 3:01 p.m.,
in response to message #19 by Massimo Gnerucci (Italy)

Quote:

It may be fun studying another language... when you have time and you can practice it everyday.

I think this is the key to the situation. I suspect high school and college students who were lucky enough to be able to afford a 48-Series machine (and were inclined towards programming in the first place) were

able to pick up RPL with no problem because they were at the point in their lives when their job was to learn.

I bought several goodies from EduCalc - like an almost brand-new 41CX and a used 71B - that they had taken as trade-ins on purchases of 48S/SX/G/GX. EduCalc really pushed the 48's. I wonder how many of those buyers regretted giving up their RPN calculators when they were having ...

Quote:

No fun at all when time is not on your side and have to remember another way to write down an if/then/else.

Re: RPN/RPL - What?

Message #21 Posted by **Johnny Billquist** on 11 June 2003, 2:18 p.m.,
in response to message #10 by Valentin Albillo

A minor nitpick: RPL is not an OOP language, but a functional language.

Re: RPN/RPL - What?

Message #22 Posted by **Jack** on 14 June 2003, 2:19 a.m.,
in response to message #10 by Valentin Albillo

Lutz - Reverse Polish Language.. not Reverse Polish Logic, Thanks, that makes more sense.

I think I get it, RPL is really "Reverse Programming Language", the hazy point in the evolution of the calculator when someone at HP declaired Hey! This is not just recording keystrokes anymore, this is a high level programming language, and it needs a name.

I don't see the more powerful instruction set or merged instructions as being non classical RPN and I don't think the infinite stack rules out RPN either, although I tend to agree with Albillo, "More than 4 levels would be a book case of diminishing returns"**. Enhancements accompanied each new model, somtimes huge enhancements. Memory increased, new commands were added, program steps became more powerful, but I don't see a point that RPN died or was replaced. RPN never was a language. It had a language though from the very start (HP-65), I just didn't know what to call it. RPL "Reverse Programing Language" works for me. The new calculator programmers with "Reverse Polish Language" RPL can call it RPN if they want, I'll just consider their RPL a newer version of my RPL. After all basic has come a long way from Dartmouth and it's still basic.

Of course I never programed with the "new RPL" so all this is easy for me to say :)

Jack

www.livingsoftware.net

**However, in Euclid's RPN I went with a 5 level stack* because I found that extra stack level often allowed even a poorly planned solution to be completed. *(ok 6 levels, counting the phantom register)

Re: RPN/RPL - What?

Message #23 Posted by **Mark Ordal** on 18 June 2003, 4:49 p.m.,
in response to message #22 by Jack

"More than 4 levels would be a book case of diminishing returns"

Perhaps this is true for just keyboard use. One strange use I found for the "unlimited" stack depth of the HP28S/48SX/48G was efficient evaluation of rational approximations for certain special functions. When the approximation polynomial is expressed in Horner's Form, you can put everything on the stack and then run through a FOR loop consisting of just *+ the right number of times. In my testing (about 14 years ago on my 48SX) the speed up was measurable compared with just "brute forcing it". (I didn't keep a record of just how much speed was gained---I just recall that it was enough of an improvement for me to consider this the preferred method.)

--Mark

"diminishing returns"

Message #24 Posted by **Euclid** on 22 June 2003, 3:29 a.m.,
in response to message #23 by Mark Ordal

Well.. with an unlimited stack you could from the keypad enter 50 numbers and then press add hmmm.. 49 times but that's not very not effcient memory wise or ease. Speed is not always everything. Ok I admit programmically, sometimes you need to step outside the box, but I remember reading a quote from Vance Packard where he said "If you had a stack of infinite levels, then you can solve an infinitely complex equation. For all practical purposes four levels is almost infinite."

I guess the point is that your weighing speed vs memory and comming up with a win, but what if you had thousands of iterations, or hundreds of thousands of iterations? Not even HP's RPL had an truly unlimited stack.

Euclid :)

Re: "diminishing returns"

Message #25 Posted by [Vassilis Prevelakis](#) on 23 June 2003, 11:46 p.m.,
in response to message #24 by Euclid

Euclid on 22 June 2003, 3:29 a.m. wrote:

> Well.. with an unlimited stack you could from the keypad enter 50 numbers and then press add hmmm..
> 49 times but that's not very not efficient memory wise or ease.

Depends, I used to do this very often with my 48SX when adding long lists of numbers. I would enter all the numbers and then confirm each one (on the Y register) before pressing +

Otherwise you have to double check numbers as you are entering them.

**vp

Re: RPN/RPL - Thanks to all

Message #26 Posted by [jimc](#) on 10 June 2003, 3:15 p.m.,
in response to message #7 by jimc

Well, as a result of the cooperative spirit and generous individuals (who continue to amaze me on this site), I have a whole bunch of new reading material for my bedside table. Who knows? Maybe this new kid on the block (my 48) will finally displace my 41. Stranger things have happened....

In case of doubts or bold attempts...

Message #27 Posted by [Vieira, Luiz C. \(Brazil\)](#) on 11 June 2003, 12:08 a.m.,
in response to message #26 by jimc

Hi, Jimc;

As a regular Brazilian-Joe, I try to be an specialist in general assumptions, so I got into a lack of expertise in almost everything I do, taking place in superficial matters.

Anyway, I offer my humble contribution in USER RPL whenever you want. Unfortunately, I have not found the time to go into SYStem RPL and Saturn assembly, but as you are begining with...

I'm sure others will support you as well, and even better than I, so you can count on us , RPL addicted (maybe I'm not speaking for all others...).

BUT, in no circumstances I'd support you if you decide to displace your HP41 for an HP48! Unless you send it to me as a gift... q8^D Kidding, guys, just kidding

Keep your HP41 and use it as well!

Best regards.

Luiz C. Vieira - Brazil

Programmining in RPL vs older calcs

Message #28 Posted by [unspellable](#) on 11 June 2003, 1:15 p.m.,
in response to message #27 by Vieira, Luiz C. (Brazil)

It has been said here that it is somehow more difficult to write a simple program in RPL than with one of the older RPN machines. Frankly, I find it much easier. RPL will handle both RPN and algebraic notation. For a very simple program I can wite it almost as quickly as directly solving the problem. I frequently write a very simple program if the calculation is to be repeated. I do a lot of calculating with complex numbers, and here RPL is vastly easier than the older HP calculator "languages".

On the whole, I find RPL to be more intuitive than the older languages. And I am not a programmer otherwise.

Perhaps part of the problem is confusion on the points that RPN is a notation, not a language or operating system, while RPL is both a language and an operating system.

[[Return to Index](#) | [Top of Index](#)]



[Go back to the main exhibit hall](#)