♥MoHPC♠ *The Museum of HP Calculators*

HP Forum Archive 21

[Return to Index | Top of Index]

My birthday, so a little commemorative mini-challenge !

*Message #1 Posted by **Valentin Albillo** on 5 Mar 2013, 7:34 a.m.*

Hi all,

As fate would have it, today's my birthday (don't ask) so I think that a little commemorative mini-challenge is in order, namely:

The commemorative mini-challenge

Write a short program or code snippet to evaluate a given polynomial at any given particular argument X, without explicitly using either multiplication/division operators (*, /)(that excludes reciprocals as well) or addition/subtraction operators (+, -).

For good measure, any and all kinds of conditional, looping, or Go-To statements or constructions are forbidden as well so no equivalents to IF, GOTO, SELECT, CASE, UNTIL, WHILE, LOOP, FOR, ... you get the point.

To simplify matters a little you can particularize your code to evaluate this particular 5th-degree polynomial:

$y = 5\ x^5 + 8\ x^4 + 4\ x^3 + 2\ x^2 + 3\ x + 1$

for any given argument X (the generalization to arbitrary polynomials is obviously straightforward).

Assorted HP models can be used for this challenge (not all can deliver) but in case your particular model has a built-in (or library's) polynomial evaluation functionality (say "CALL POLYEV(coefficients, X)") you're asked **not** to use it lest you'll ruin the challenge for yourself.

I'll give a 3-line HP-71B solution, which runs like this:

```
    >RUN
      ? PI                   ( X value )
```

```
            2463.56024568    ( P(x)    )
    >RUN
      ? -7
            -66121
```

Try your hand at it, it's actually quite easy !

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #2 Posted by ***Gerson W. Barbosa*** *on 5 Mar 2013, 9:38 a.m.,*
in response to message #1 by Valentin Albillo

Hola Valentin,

First of all, ¡feliz cumpleaños!

I am not sure whether this is what you had in mind, but it does what you want if exponentiation is allowed.

```
PLEVAL:
%%HP: T(3)A(D)F(,);
\<< { 5, 4, 3, 2, 1, 0, } ^ OBJ\-> \->ARRY [ 5, 8, 4, 2, 3, 1, ] DOT
\>>
```

On my hp 50g:

```
pi PLEVAL  -->  2463.56024568
-7 PLEVAL  -->        -66121.
```

Best regards,

Gerson.

P.S.: This solves your example only, so this is not a solution.

Edited: 5 Mar 2013, 9:49 a.m.

Re: My birthday, so a little commemorative mini-challenge !

*Message #3 Posted by **Gilles Carpentier** on 5 Mar 2013, 12:20 p.m.,*
in response to message #2 by Gerson W. Barbosa

Hi Gerson

A little improvment :

```
\<< { 5 4 3 2 1 0 } ^ AXL [ 5 8 4 2 3 1 ] DOT \>>
```

Happy birthday Valentin ;)

Edited: 5 Mar 2013, 12:23 p.m.

Re: My birthday, so a little commemorative mini-challenge !

*Message #4 Posted by **Gerson W. Barbosa** on 5 Mar 2013, 12:35 p.m.,*
in response to message #3 by Gilles Carpentier

Hi Gilles,

Thank you! I had never heard of AXL (array <-> list) before. This was new to the 49G series, as I've found out in the Advanced User's Guide.

Gerson.

Edited: 5 Mar 2013, 12:38 p.m.

Re: My birthday, so a little commemorative mini-challenge !

*Message #5 Posted by **Gerson W. Barbosa** on 5 Mar 2013, 3:57 p.m.,*
in response to message #3 by Gilles Carpentier

I guess SEQ belongs in the looping category, thus forbidden. Also, using LN EXPM to subtract 1 might be cheating. Anyway, here is my generalization attempt:

```
PLYEVAL:
%%HP: T(3)A(D)F(.);
\<< SWAP DUP SIZE OBJ\-> DROP LN EXPM IP 'X' 0. OVER SWAP 4. ROLL 1. SEQ REVLIST ROT SWAP ^ AXL DOT
\>>
[ 5 8 4 2 3 1 ] pi PLYEVAL  -->  2463.56024568
[ 5 8 4 2 3 1 ] -7 PLYEVAL  -->        -66121.
     [ 1 -5 6 ]  2 PLYEVAL  -->             0.
```

Edited to fix a typo.

Edited: 5 Mar 2013, 4:16 p.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #6 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:29 a.m.,*
in response to message #3 by Gilles Carpentier

Hi, Gilles:

Thank you very much. Nice, elegant, short solution, by the way, just what I had in mind (see my original solution below).

I knew that it was easy but expected it to last a few hours and survive a few attempts ... :-D

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #7 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:26 a.m.,*
in response to message #2 by Gerson W. Barbosa

Hi, Gerson !

Thank you very much for your "felicitación", much appreciated.

And yes, your solution (which it actually *is*, as I stated that evaluating the specific polynomial was sufficient) is just what I had in mind, see my original solution below.

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #8 Posted by ***Thomas Klemm*** *on 5 Mar 2013, 10:19 a.m.,* *in response to message #1 by Valentin Albillo*

Hi Valentin

Congratulations!

Here's a solution for the HP-42S:

```
00 { 40-Byte Prgm }
01 STO 00
02 \GSREG 01
03 CL\GS
04 5
05 Y^X
06 5
07 \GS+
08 RCL 00
09 4
10 Y^X
11 8
12 \GS+
13 RCL 00
14 3
15 Y^X
16 4
17 \GS+
18 RCL 00
19 2
20 Y^X
21 2
22 \GS+
23 RCL 00
24 3
25 \GS+
26 1
27 ENTER
28 \GS+
29 RCL 05
30 END
```

Not sure whether using \GS+ is allowed. I considered using storage arithmetic kind of cheating.

Kind regards
Thomas

Edited: 5 Mar 2013, 10:24 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #9 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:33 a.m.,*
in response to message #8 by Thomas Klemm

Hi, Thomas:

Thanks a lot for your congratulations, much appreciated.

Your attempt is nice and short but regrettably it uses the GS+ which, having a "+" sign and being essentially an explicit sum (much as STO+, etc), is not allowed per the mini-challenge requirements.

But thanks anyway for submitting it and for your interest.

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #10 Posted by ***Walter B*** *on 5 Mar 2013, 10:24 a.m.,*
in response to message #1 by Valentin Albillo

Feliz cumpleaños, Valentin!

I don't want to spoil your challenge - just want to make sure I got it right: we are allowed to use every calculator key except [+], [-], [*], [/], and [1/x], aren't we?

d:-I

Re: My birthday, so a little commemorative mini-challenge !

Message #11 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:35 a.m.,*
in response to message #10 by Walter B

Hi, Walter:

Quote:

> Feliz cumpleaños, Valentin!
>
> I don't want to spoil your challenge - just want to make sure I got it right: we are allowed to use every calculator key except [+], [-], [*], [/], and [1/x], aren't we?
>
> d:-I

Thank you very much for your kind congratulations, in mi native Spanish no less, what a luxury !! ... :-)

As per your question, yes, the explicit symbol operators for the four arithmetic operations (plus reciprocals) aren't allowed.

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #12 Posted by ***Eddie W. Shore*** *on 5 Mar 2013, 3:42 p.m.,*
in response to message #1 by Valentin Albillo

Happy Birthday Valentin! :) May it be a good one.

Re: My birthday, so a little commemorative mini-challenge !

Message #13 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:38 a.m.,*
in response to message #12 by Eddie W. Shore

Hi, Eddie !

Thank you very much for your kind congratulation.

And yes, it was a perfectly fine birthday, with cake and little candles and all that jazz ... :-)

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

*Message #14 Posted by **Massimo Gnerucci (Italy)** on 5 Mar 2013, 5:22 p.m.,*
in response to message #1 by Valentin Albillo

Happy birthday, Valentin! :)

Re: My birthday, so a little commemorative mini-challenge !

*Message #15 Posted by **Valentin Albillo** on 6 Mar 2013, 4:41 a.m.,*
in response to message #14 by Massimo Gnerucci (Italy)

Thanks a lot, Massimo, much appreciated.

Now to begin the **2**nd half of my **6**th decade ... (2,6 -> 26 = **13** + **13**).

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

*Message #16 Posted by **Gilles Carpentier** on 5 Mar 2013, 5:43 p.m.,*
in response to message #1 by Valentin Albillo

A generic solution 65 Bytes for 50G

```
« OVER AXL 1 « DROP ENDSUB  NSUB - » DOSUBS ^ AXL DOT »
```

entry is for example [5 8 4 2 3 1] -7 But I use a - ....

Re: My birthday, so a little commemorative mini-challenge !

*Message #17 Posted by **Valentin Albillo** on 6 Mar 2013, 4:43 a.m.,*
in response to message #16 by Gilles Carpentier

> Quote:
>
> ---
>
> ```
> But I use a -
> ```

> ....

Tsk, tsk ... nasty boy ! ... :-D

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #18 Posted by ***Gerson W. Barbosa*** *on 11 Mar 2013, 6:18 p.m.,*
in response to message #16 by Gilles Carpentier

> Quote:
>
> ```
> « OVER AXL 1 « DROP ENDSUB  NSUB - » DOSUBS ^ AXL DOT »
> ```

Very nice and elegant!

Better ignore the plus, minus & times restrictions, now that the mini-challenge is over. The generic solution below is 111 bytes long, but avoids exponentiation (should be faster and more accurate for higher order polynomials). It looks a bit clumsy, however. Any idea for a shorter, faster and elegant solution that uses no exponentiation?

```
%%HP: T(3)A(D)F(,);
\<< OVER AXL DUP / *
DEPTH \-> d
  \<< 1 { 1 } REPL 1
SWAP
    \<< * DUP
    \>> STREAM DEPTH
SWAP DROP d - \->LIST
  \>> REVLIST AXL DOT
\>>

------------------------------

%%HP: T(3)A(D)F(,);
\<< TIME
[ 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 2 3 5 5 7 8 1 7 3 55 8 4 2 3 1 ]
```

```
\pi PLEV TIME ROT
HMS-
\>>
```

This returns 1.01465154025E17 in 2.14 seconds on my HP-48GX (with Erable & Metakernel), only slightly faster than your size-optimized code (2.33 seconds). Built-in PEVAL is about eight times faster.

Edited: 11 Mar 2013, 10:36 p.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #19 Posted by ***Csaba Tizedes (Hungary)*** *on 6 Mar 2013, 3:49 a.m.,*
in response to message #1 by Valentin Albillo

Hi Valentin,

Happy Birthday to You! :)

So, we livin in a material world, therefore my solution according to this:

```
1.) Grab your favourite Financial calc, clear the CF regs
2.) Type x, 'ENTER', type '1' then 'delta%' and store the result in 'i'
3.) Put the polynom coefficients into a CF list: 1 CF0, 3 CF1, 2 CF2, etc...
4.) Calculate Net Present Value
```

This works x=PI, but not for x=-7... :(

Csaba

Re: My birthday, so a little commemorative mini-challenge !

Message #20 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:52 a.m.,*
in response to message #19 by Csaba Tizedes (Hungary)

Hi, Csaba:

Thanks a lot for your congratulations, much appreciated.

Very nice solution, indeed. Re your technique, I actually wrote many years ago an article on generic Nth-degree polynomial root finding and evaluation in the HP financial model *par excellence*, the HP-12C, namely:

HP-12C Serendipitous Solver

Congratulations to you for using this little-known capability, well done !

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #21 Posted by ***Paul Dale*** *on 6 Mar 2013, 4:40 a.m.,*
in response to message #1 by Valentin Albillo

Nice problem. How to get multiplies and adds. Rectangular to polar conversion gives sqrt($x^2 + y^2$). With lots of squaring and square roots this will give a kind of addition and logs and exponentials would produce multiplication. However, it won't handle negative numbers properly.

Another approach then. Complex multiplication would work but that is a multiply which isn't permitted. How about a dot product instead? That does work. A WP-34S solution is based around this sequence which performs one step in the Horner expansion. It effectively calculates sum*x + B with the current sum in X and the x value in L.

```
    #001
    x[<->} Y
    B
    RCL L
    [cmplx]DOT
```

Then an altered first sequence and any polynomial can be done. The example quintic follows:

```
001 LBL A
002 # 001
003 # 005
004 # 008
005 RCL T
006 [cmplx]DOT
007 # 001
008 x[<->} Y
009 # 004
010 RCL L
011 [cmplx]DOT
012 # 001
013 x[<->] Y
014 # 002
015 RCL L
```

```
016 [cmplx]DOT
017 # 001
018 x[<->] Y
019 # 003
020 RCL L
021 [cmplx]DOT
022 # 001
023 x[<->] Y
024 # 001
025 RCL L
026 [cmplx]DOT
027 END
```

All the short integer constants aren't required, although steps 2 through 4 need to be separated with ENTER if not.

With loops and coefficients stored in registers this would be a lot more compact.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #22 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 4:56 a.m.,*
in response to message #21 by Paul Dale

Hi, Paul:

Brave attempt, which stumbles upon the right idea (using DOT) and implements it as well as it can without the recourse to vector or matrix operations, which I understand aren't available in this particular model, nor loops, which are disallowed by the mini-challenge's requirements.

Well done.

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #23 Posted by ***Paul Dale*** *on 6 Mar 2013, 5:05 a.m.,*
in response to message #22 by Valentin Albillo

The 34S does have matrix operations, although they are a little clumsy. Remember those really ill conditioned matrices you presented last year?

Unfortunately, all the relevant matrix operations include + or * in their names. That leaves us with the LU decomposition, determinant and inverse. The library has a few more: trace & absolute trace, row and column norms and the Frobenius norm. It might be possible to solve this challenge using these.

There are also some 3D vector routines in the library but only the dot product is usable without encroaching on the no arithmetic operators rule. This isn't really different to the solution above.

I'm also loathe to include the library functions in solutions.

- Pauli

Edited: 6 Mar 2013, 5:54 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #24 Posted by ***Paul Dale*** *on 6 Mar 2013, 4:57 a.m.,*
in response to message #1 by Valentin Albillo

Another thought. On the 34S we have a good selection of percentage calculations.

Specifically:

- % returns (x * y) / 100
- %T returns 100 * x / y
- %MG returns 100 * (x-y) / x

The first allows multiplication of x and y. It also allows a rescale by 0.01 if either x or y is unity.

The second gives division which we don't need here. More importantly, it allows multiplication by 100 if y is 1. Combining this with the above gives us straight out multiplication. A * B is calculated with a sequence like this:

```
A
B
%
1
x<>y
%T
```

The third gives us subtraction when combined with the first, providing A is not zero. If A is zero, we've got CHS that does the same job. A - B is calculated via:

```
B
A
%MG
Last X
%
```

At this point any polynomial can be coded via Horner's method, albeit rather painfully.

- Pauli

Edited: 6 Mar 2013, 5:06 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #25 Posted by ***Paul Dale*** *on 6 Mar 2013, 5:17 a.m.,*
in response to message #24 by Paul Dale

Extending this line of thought and using percent change it is possible to get a sequence that works on a 12C. The scientific models really are left behind this all-star calculator.

Remember the formula for percent change:

- Delta% returns 100 * (x-y) / y

This can be used instead of %MG above provided y is not zero. If y is zero, then we don't need to subtract anything and can skip that step.

The sequence to get A - B is therefore:

```
B
ENTER
A
Delta %
%
```

Combine this with the % and %T multiplications and Horner's method and again any polynomial can be evaluated subject to the challenge's limitations.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #26 Posted by ***Paul Dale*** *on 6 Mar 2013, 5:47 a.m.,* *in response to message #25 by Paul Dale*

To prove the 12C can solve this one:

```
        01      STO 0
        02      5
        03      %
        04      1
        05      x<>y
        06      %T
        07      FRAC
        08      8
        09      CHS
        10      ENTER
        11      LST X
        12      delta%
        13      %
        14      RCL 0
        15      %
        16      1
        17      x<>y
        18      %T
        19      FRAC
        20      4
        21      CHS
        22      ENTER
        23      LST X
        24      delta%
        25      %
        26      RCL 0
        27      %
        28      1
        29      x<>y
        30      %T
        31      FRAC
        32      2
        33      CHS
        34      ENTER
        35      LST X
        36      delta%
        37      %
        38      RCL 0
        39      %
        40      1
        41      x<>y
        42      %T
        43      FRAC
```

```
44      3
45      CHS
46      ENTER
47      LST X
48      delta%
49      %
50      RCL 0
51      %
52      1
53      x<>y
54      %T
55      FRAC
56      1
57      CHS
58      ENTER
59      LST X
60      delta%
61      %
```

I suspect that with a lot more pain, the use of the register could be avoided as well. I don't know if the 12C has enough memory to hold the resulting program however.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #27 Posted by ***Paul Dale*** *on 6 Mar 2013, 5:23 a.m.,*
in response to message #1 by Valentin Albillo

Happy birthday and thanks for this interesting mini-challenge.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #28 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 6:07 a.m.,*
in response to message #27 by Paul Dale

Hi again, Paul:

Thank you very much for your congratulations, I really appreciate it.

Also, thanks for your interest in this mini-challenge and for your posted attempts, which are of great interest themselves. As I think the 34s does include full financial functions, they can be used as well to solve it, as explained in the posts above.

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #29 Posted by ***Paul Dale*** *on 6 Mar 2013, 6:34 a.m.,*
in response to message #28 by Valentin Albillo

The 34S doesn't have full financial functions. It is deliberately lacking here, we never intended it to be a financial calculator. It does, however, have a very good selection of percentage calculations and a robust TVM solver in the standard library.

The fact that a 12C can solve this challenge is great and actually caught me a bit by surprise. Never underestimate where arithmetic operations are hiding in other commands.

I suspect the least featured scientific calculator that can solve this challenge is the 15C. At least I can see how to do it on the 15C using MATRIX 6 and I can't (yet) see how to do it on a 12C or 32S. I'd love to be proven incorrect here.

Some possibilities to consider remain:

- Polar/Rectangular conversions aren't quite enough. They can add absolute values with some trickery and multiply absolute values with logarithms, exponentials and the add.
- Unit conversions are too limited to be generally useful. Some multiplications and temperature conversions can add as well but they aren't flexible enough.
- Integration might provide something workable but it probably falls afoul of the no branches -- although the 32Sii can integrate a forumla so that one might pass. The integral from 0 to x of y dt gives the product x*y. The integral from x to y of 1 dt gives y-x. I guess this is a possibility that could be followed -- subject to clearing up if these are even allowed under the rules.
- At this point I'm out of obvious or easy to use / disprove ones.
- The 34S has some orthogonal polynomials included which just might be workable but I think it would be quite difficult to massage them into shape.
- The 34S also has the product and sum commands, again they are branching and are tricky to set up but should be able to solve this problem.

- I also thought about integer mode solutions. Knowing full well that the PI example would have been impossible. I suspect that a solution not involving conditional tests will be very difficult.

Still, I'm sure other people will come up with some more intriguing possibilities.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #30 Posted by ***Paul Dale*** *on 6 Mar 2013, 6:46 a.m.,*
in response to message #29 by Paul Dale

Maybe an integer mode solution is a possibly. Addition can be done via basic logical operations xor, or and and plus some shifts or rotates.

From this multiplication by a constant is possible via repeated additions. You know what the constant is so you can inline that many additions. There are better algorithms but I don't see how to implement them branchlessly.

It would be truly horrible without subroutines or loops but I think it is theoretically possible.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #31 Posted by ***Gerson W. Barbosa*** *on 6 Mar 2013, 6:53 a.m.,*
in response to message #30 by Paul Dale

> Quote:
>
> Addition can be done via basic logical operations xor, or and and plus some shifts or rotates.

I had considered R->B #1h XOR B->R instead of LN EXPM IP on the HP 50g, but I quit because the former requires more bytes.

Gerson.

Edited: 6 Mar 2013, 6:54 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #32 Posted by ***Paul Dale*** *on 6 Mar 2013, 7:12 a.m.,* *in response to message #31 by Gerson W. Barbosa*

The integer code sequence is wrong. It converts 2 to 3 and 3 back to 2. It isn't counting. I do like LN EXPM to count.

I'm sure there are sequences of logical operations that will do the add. RJ justifies the number and returns a count of how many zeros it skipped over. That could then be used as an argument to a mask or shift command to get a decrement operation. You want to set the lowest 1 bit to zero and all zero bits below that to one.

Something like this sequence will subtract one:

```
    RJ
    x<>y
    1
    XOR
    SL->Y
    MASKR->Y
    OR
```

I suspect a parallel adder is a possibility and would be shorter than a sequence of these.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #33 Posted by ***Gerson W. Barbosa*** *on 6 Mar 2013, 8:20 a.m.,*
in response to message #32 by Paul Dale

> Quote:
>
> The integer code sequence is wrong. It converts 2 to 3 and 3 back to 2.

Same on the the HP-42S:

```
1 3 XOR --> 2
2 1 XOR --> 3
```

It's obvious now only odd numbers will be subtracted one unit this way.
On the HP-42S, NOT ABS is equivalent to 1 + (or INC X on the wp34s).

Gerson.

Edited: 6 Mar 2013, 8:21 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #34 Posted by ***Paul Dale*** *on 6 Mar 2013, 3:55 p.m.,*
in response to message #33 by Gerson W. Barbosa

That is a nice little subtract one sequence. Only works in two's complement or unsigned modes and doesn't work for negative numbers.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #35 Posted by ***Gerson W. Barbosa*** *on 6 Mar 2013, 11:26 p.m.,*
in response to message #34 by Paul Dale

NOT +/- will always add one to integers. Useful only to overcome no + restrictions, I think.

Gerson.

Re: My birthday, so a little commemorative mini-challenge !

Message #36 Posted by ***Paul Dale*** *on 7 Mar 2013, 12:51 a.m.,*
in response to message #35 by Gerson W. Barbosa

This works for twos complement and unsigned, but not the other two integer sign modes on the 34S or ones complement on the 16C. Still it is a nice way to avoid +.

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

Message #37 Posted by ***Walter B*** *on 6 Mar 2013, 6:44 a.m.,*
in response to message #28 by Valentin Albillo

> Quote:
>
> ... I think the 34s does include full financial functions ...

FYI, it only contains a small subset. There is no lack of reasonable financial calculators in this world (maybe there's a lack of reasonable persons consciously operating them but that's another topic) so we focussed on scientific functions. What Pauli mentioned is pretty much all of the %-functions supported. If you want to know more about it, please look here (8.7MB) or get the book.

d:-)

Re: My birthday, so a little commemorative mini-challenge !

*Message #38 Posted by **Paul Dale** on 6 Mar 2013, 6:58 a.m.,*
in response to message #37 by Walter B

Not even close to them all :-) There are three more:

%	x * y / 100
%MG	100 * (x-y) / x
%MRR	100 * $((x/y)^{1/z} - 1)$
%T	100 * x / y
% Sigma	100 * x / Sigma X
%+MG	y / (1 - x / 100)
Delta %	100 * (x-y) / y

In an earlier iteration we also had:

%+	y + x * y / 100
%-	y - x * y / 100

- Pauli

Re: My birthday, so a little commemorative mini-challenge !

*Message #39 Posted by **Walter B** on 6 Mar 2013, 9:11 a.m.,*
in response to message #38 by Paul Dale

Ok, ok, I stand corrected :-) Still I think [Delta]% and %MRR are the only two really useful, maybe also the margin calculations to some extent. The other ones I'd do faster manually than looking them up in X.FCN.

d:-)

Re: My birthday, so a little commemorative mini-challenge !

Message #40 Posted by ***Valentin Albillo*** *on 6 Mar 2013, 6:36 a.m.,*
in response to message #1 by Valentin Albillo

Hi to all:

First of all, thanks a lot to all of your for your interest and clever solutions to this commemorative mini-challenge.

My own original solution is the following 3 lines of HP-71 B code:

```
    1  DESTROY ALL @ OPTION BASE 0 @ DIM A(5),B(5) @ READ A
    2  DATA 5,8,4,2,3,1,x^5,x^4,x^3,x^2,x,1
    3  INPUT X @ READ B @ PRINT DOT(A,B)


    >RUN


      ? PI
            2463.56024568
    >RUN


      ? -7
           -66121
```

A few comments:

- This solution evaluates the particular polynomial I gave, but it's easily generalized to arbitrary polynomials
- it uses READ to both fill-in the coefficients array and to actually evaluate and fill a vector with the required powers of X, including X^0 (=1)
- it then uses DOT (dot product) to non-iteratively compute in one go both the multiplication of each power of X times the respective coefficient, and the sum of all partial products to give the final value for the polynomial expression. A single DOT does it all.
- once the coefficients vector has been filled in (READ A) or input in some way (MAT INPUT A, for instance), each evaluation of P(x) at a particular X can be done by inputting X, READing (which evaluates) again the powers of said X into the powers vector, and using DOT with both vectors as arguments. No need to re-read the coefficients, which are left unaltered by the evaluation process.
- outside of this mini-challenge, this technique can also be generalized to evaluate an arbitrary number of polynomials at a particular X value in one go. You just dimension A to be a matrix, read all the coefficients for the various polynomials into this matrix, then read (evaluate) the required powers of X into a vector B, and finally execute a single matrix product statement (MAT Y=A*B) to fill vector Y with all the values of the polynomials evaluated at X.

Essentially you get all polynomials evaluated at a given X with just a single READ and a single MAT* statement. Notice that the number of polynomials is arbitrary and that they can be of different degrees, just input the leading coefficients as 0 as necessary.

Simple and elegant, isn't it ? ... :-)

That's all. Thanks for your interest and

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #41 Posted by ***aurelio*** *on 6 Mar 2013, 8:10 a.m.,*
in response to message #40 by Valentin Albillo

Happy birthday Valentino excuse me for the late and thank you for your challenge idea, it has been a chance for me to take one more lesson in programming technique by the "GOTHA" in this matter , here on this wonderful board.

Aurelio

Edited: 6 Mar 2013, 8:12 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #42 Posted by ***Valentin Albillo*** *on 7 Mar 2013, 3:30 a.m.,*
in response to message #41 by aurelio

Thanks a lot, **aurelio**.

I'm glad you've found my commemorative mini-challenge interesting and perhaps you might consider the techniques discussed as an useful addition to your programming knowledge chest.

Best regards from V.

Re: My birthday, so a little commemorative mini-challenge !

Message #43 Posted by ***nina scholz*** *on 7 Mar 2013, 3:30 a.m.,*
in response to message #40 by Valentin Albillo

```
LBL A
STO 8
CLEAR Sigma
RCL 8
5
y^x
5
Sigma+
RCL 8
4
y^x
8
Sigma+
RCL 8
3
y^x
4
Sigma+
RCL 8
x^2
2
Sigma+
RCL 8
3
Sigma+
1
ENTER
Sigma+
RCL 7
```

Edited: 7 Mar 2013, 3:33 a.m.

Re: My birthday, so a little commemorative mini-challenge !

Message #44 Posted by ***Walter B*** *on 7 Mar 2013, 3:44 a.m.,*
in response to message #43 by nina scholz

Please see this post above.