[VA] SRC #007 - 2020 April 1st Ramblings

04-01-2020, 08:52 PM **Post: #1**

Valentin Albillo
Senior Member

Posts: 495
Joined: Feb 2015
Warning Level: 0%

[VA] SRC #007 - 2020 April 1st Ramblings

Hi all, welcome to my new ***SRC #007 - 2020 April 1st Ramblings***:

As usual, every time April 1^{st} comes by I tend to indulge in assorted math/computing ramblings, 9 of which I'll presently share with you. Also, you surely own a favorite HP calc which you're proud to show off to everyone and their uncle, and you have no trouble solving whatever math problems or computing tasks come your way using it, so this will be an ideal opportunity to put it to work for good. Let's begin:

__Note__: In what follows, __[x]__ is the integer part of x, __{x}__ is the fractional part of x, and __Ceil(x)__ is the ceiling of x, e.g.: [3.14] = 3, {3.14} = 0.14, Ceil(3.14) = 4.

1) Solving a system of *N* plain-vanilla linear equations in *N* unknowns is dead easy with most HP calculators but as soon as you introduce some very minor changes things aren't that easy anymore. For instance I wonder what the *solution* is for this system:

```
 u  + [v] + {w} = 200.0
{u} +  v  + [w] = 190.1
[u] + {v} +  w  = 178.8
```

2) In the same vein, we're used to the fact that a plain-vanilla 2^{nd}-degree equation has exactly two roots, real or complex, and our HP calcs have no problem finding them. But I wonder about the *real roots* of this slightly-modified "2^{nd}-degree" equation (how many roots, their values ...):

$$x^2 - 10\,[x] + 12.75 = 0$$

3) Now, here's another 2^{nd}-degree equation in *X*:

$$X^2 + 2\,X + 5\,I = 0$$

with the caveat that this time *X* is not just some scalar value but a square *matrix* and *I* is the corresponding *Identity* matrix. Matricial equations can have any number of roots, including an infinity of them or none at all, and it would be nice to find some *roots* for this equation, if they do exist.

4) Also, after dealing with finding some roots of the above matricial equation I then considered the converse problem, i.e.: to find an N^{th}-degree equation which has a given *NxN* square matrix as a root. For instance, I wonder what 3^{rd}-degree matricial equation (if any) would have the following 3x3 matrix as a root:

```
 2   3   5
 7  11  13
17  19  23
```

5) The N^{th} *Fibonacci* number (belonging to the well-known *Fibonacci sequence: 1, 1, 2, 3, 5, 8, 13 ...*) is given by an expression involving one or two exponential functions a^N, where *a* is the *Golden Ratio*, 1.618+. Since the *hyperbolic* *functions* are also combinations of exponential functions e^x, where *e* is 2.718+, I wonder if there's some simple way to express the N^{th} *Fibonacci* number using them, perhaps as simple as a few lines of *RPN/RPL* code or a short single-line user-defined function, used like this:

FNF(1) = 1, FNF(2) = 1, FNF(3) = 2, ..., FNF(10) = 55, ...

6) The following expression (where *N > 0* is an *integer* and *log2* is the natural logarithm of 2 = 0.693+):

$\text{Ceil}(2/(2^{1/N} - 1)) = [2*N/\log2]$

seems to be an *identity* for *all* integer values of n. For instance for *N = 1* both sides equal *2*, for *N = 5* both sides equal *14*, and for *N = 2020* both sides equal *5828*. Now I wonder if there are any *exceptions* at all and, if yes, whether a short program (say 4 lines of code) would be able to very quickly find *the first three*.

7) I'm a fan of *Star Trek* since always and yesterday I came up with the following hypothetical, idealized scenario (assume all times and distances are *exact* and *lightspeed* is exactly 300,000 Km per second):

Captain Kirk is in command of the *USS Enterprise* and has to carry out a most critical mission: an unstable star is going supernova exactly 6 weeks from the present moment, destroying several inhabited planets and causing massive loss of life. However, the explosion can be inhibited using certain exotic ore available in some other nearby uninhabited system located just 504 lighthours away, so the *USS Enterprise* is ordered to travel there, beam the ore onboard, then return to the unstable star and use the ore to stop the supernova explosion from ever happening.

Now, the *USS Enterprise* can use two different types on engine. The main one is the *warp engine*, which is used for interstellar travel and can achieve faster-than-light speeds depending on the *cube* of the warp factor engaged, i.e., warp 2 achieves 2^3 = 8x lightspeed, while warp 10 achieves 10^3 = 1,000x lightspeed. The other engine type is the *impulse engine*, which is used while orbiting some planet or coasting to a starbase and other such relatively low-speed navigation as well as for emergencies, achieving all speeds from 0 at rest to just shy of lightspeed at *full impulse*.

As (pretty bad) luck would have it, the very moment the *USS Enterprise* starts her journey to reach the ore the warp engine suddenly fails utterly and further Scotty reports that without the help of the warp field the impulse engine can only achieve *half impulse* (= 150,000 Km per second) so that's what Kirk orders and the *Enterprise* goes for the ore at half impulse while warp engine repairs are underway on the double.

Nevertheless, despite the unexpected setback Kirk's not especially worried as he knows his starship is capable of sustaining speeds up to *warp 14* (2,744x lightspeed) if needed for the return leg of the journey and fortunately by the time the *USS Enterprise* arrives at the ore and beams it aboard (which takes no significant time, assume instantly) the warp engine is back to fully operational status and thus Kirk sets for immediate return to the unstable star to try and arrive still within the inescapable deadline.

That said, the question I pondered is this: What is the *minimum* warp factor Kirk should engage in order to meet the deadline and avoid massive loss of life ?

8) And speaking of Kirk, *Picard's Little Theorem* says that there's *at most one* value which an *entire* function does not assume. For instance, *exp(z)* is analytic in the whole complex plane and so is representable by an everywhere convergent power series, thus it's an entire function and the only value it omits is ***0*** because *exp(z)* is never *0* for any finite complex argument *z*, so it complies with *Picard's LT* alright. So far so good.

Now consider the function *exp(exp(z))*, which is also an entire function, with a power series convergent everywhere and all that jazz. Being of the form *exp(something)* it omits ***0*** because *exp(z)* does. But then it also omits ***1*** (= *exp(0)*), because its argument is *exp(z)*, which omits *0*. Thus there are *two* values it doesn't take (***0*** and ***1***), contradicting *Picard's LT*. So I wonder: what gives ?

9) As a bonus, a final spread of misc ramblings o'mine:

- the equation *Gamma(x) = Gamma(y)* has trivial solutions *x = y* but surely nontrivial solutions can be found as well ?

- the Gamma function grows faster than the exponential function, and I wonder at what positive value of *x* do their graphics cross for the last time.

- at such value you'll have *Gamma(x) = exp(x)* for some *x* and at some other related value you'll have *Gamma(Log(y)) = y* for some *y*.

- *Gamma(Pi)* and *Gamma(-Pi/2)* are surprisingly close when rounded to *2 decimal places* ...

- ... but slightly changing *Pi* to $Pi+(2/5^2)^2$ above makes both expressions agree to no less than *8 places* when truncated.

- the equation *sin(x)+(7*x+1)*cos(x+1) = x* has a surprising root in the interval [12, 14].

- and to top it all, some nice square roots:

Sqrt(95888) = 309.657875727390470000000975517...

Sqrt(22008840) = 4691.358**0123456789**961013...

Sqrt(40850970) = 6391.476355271917924894369779931786970954485049271560920714895416366380316278765517444285758541109670509316359483877**3141592654**6989...

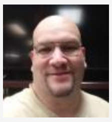
I'll eventually post some additional comments on all 9 ramblings above but let's see your comments first (if any).

Have a nice week and, if you're keeping confinement, I hope this humble effort of mine will offer you some hopefully welcome diversion for a little while. Take care !
V.

Find All My HP-related Materials here: **Valentin Albillo's HP Collection**

PM WWW FIND EDIT QUOTE REPORT

04-01-2020, 09:36 PM **Post: #2**



Eddie W. Shore
Senior Member

Posts: 1,085
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

I'm grateful for the diversions, Valentin. Live long and prosper.

EMAIL PM WWW FIND QUOTE REPORT

04-02-2020, 01:49 AM (This post was last modified: 04-03-2020 02:38 AM by Albert Chan.) **Post: #3**

Albert Chan
Senior Member

Posts: 874
Joined: Jul 2018

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Valentin Albillo Wrote:** (04-01-2020 08:52 PM)
>
> **1)** Solving a system of *N* plain-vanilla linear equations in *N* unknowns is dead easy with most HP calculators but as soon as you introduce some very minor changes things aren't that easy anymore. For instance I wonder what the *solution* is for this system:
>
> ```
>  u  + [v] + {w} = 200.0
> {u} +  v  + [w] = 190.1
> [u] + {v} +  w  = 178.8
> ```

Add them all, and halved it, u+v+w = 568.9/2 = 284.45 → {u}+{v}+{w} = .45, 1.45, 2.45
Assume {w}=0, eqn1 → {u}=0, eqn2 → {v}=.1, sum of fractional part does not match any of above.

Thus, eqn1 → {u}+{w} = 1 → {v} = 1.45-1 = .45
→ {u} = {190.1 - .45} = .65
→ {w} = 1 - {u} = .35

Removing all [], {}:

u + v = 200.0 + {v} - {w} = 200.1
v + w = 190.1 - {u} + {w} = 189.8
u + w = 178.8 + {u} - {v} = 179.0

→ (u, v, w) = (u+v+w) - (v+w, u+w, u+v) = 284.45 - (189.8, 179.0, 200.1) = (94.65, 105.45, 84.35)

EMAIL PM FIND QUOTE REPORT

04-02-2020, 02:41 AM **Post: #4**

Paul Dale

Posts: 1,582
Joined: Dec 2013

Senior Member

RE: [VA] SRC #007 - 2020 April 1st Ramblings

#7: Distance to planet is 504 light hours. At half impulse this is 1008 hours of travelling time. There are 168 hours in a week. 1008/168 = 6 weeks exactly. Infinite warp speed would be required to get back and save the doomed system.

Kirk might as well hang around and seduce the inevitably beautiful local alien women ore miners. Except that the system containing the ore is uninhabited. He'll instead seduce the also inevitable beautiful but doomed women for just under six weeks...

PM FIND QUOTE REPORT

04-02-2020, 04:10 AM **Post: #5**

Albert Chan
Senior Member

Posts: 874
Joined: Jul 2018

RE: [VA] SRC #007 - 2020 April 1st Ramblings

#2: $x^2 - 10\,[x] + 12.75 = 0$

$(\{x\} + [x])^2 - 10\,[x] + 12.75 = \{x\}^2 + 2\,[x]\,\{x\} + ([x]^2 - 10\,[x] + 12.75) = 0$

$\{x\} = -[x] \pm \sqrt{[x]^2 - ([x]^2 - 10\,[x] + 12.75)} = -[x] \pm \sqrt{10\,[x] - 12.75}$

From the radical, to have real $\{x\}$, $[x] \geq 2$

[x]	$\{x\} = \sqrt{10\,[x] - 12.75} - [x]$
2	$\sqrt{7.25} - 2 \approx 0.6926$ OK
3	$\sqrt{17.25} - 3 \approx 1.1533$ BAD
4	$\sqrt{27.25} - 4 \approx 1.2202$ BAD
5	$\sqrt{37.25} - 5 \approx 1.1033$ BAD
6	$\sqrt{47.25} - 6 \approx 0.8739$ OK
7	$\sqrt{57.25} - 7 \approx 0.5664$ OK
8	$\sqrt{67.25} - 8 \approx 0.2006$ OK
9	$\sqrt{77.25} - 9 \approx -.2108$ BAD

All bigger [x] will produce negative {x}, thus no more roots

$\rightarrow$ 4 roots = $\sqrt{7.25}$, $\sqrt{47.25}$, $\sqrt{57.25}$, $\sqrt{67.25}$

EMAIL PM FIND QUOTE REPORT

04-02-2020, 09:38 AM **Post: #6**

J-F Garnier
Senior Member

Posts: 381
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Valentin Albillo Wrote:** ⇒ (04-01-2020 08:52 PM)
>
> **8)**
> ...
> Now consider the function *exp(exp(z))*, which is also an entire function, with a power series convergent everywhere and all that jazz. Being of the form *exp(something)* it omits ***0*** because *exp(z)* does. But then it also omits ***1*** (= *exp(0)*)... contradicting *Picard's LT*. So I wonder: what gives ?

Of course, it doesn't omit ***1*** :
let's z=log(i*2*Pi*n) , n<>0
exp(exp(z)) = exp(i*2*Pi*n) = ***1***

Picard's Little Theorem is safe :-)

J-F

EMAIL PM WWW FIND QUOTE REPORT

04-02-2020, 05:20 PM **Post: #7**

Bernd Grubert
Member

Posts: 84
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

Hi Valentin,
thanks for the interesting excursion.

> **Valentin Albillo Wrote:** ⇨ (04-01-2020 08:52 PM)
>
> **1)** Solving a system of *N* plain-vanilla linear equations in *N* unknowns is dead easy with most HP calculators but as soon as you introduce some very minor changes things aren't that easy anymore. For instance I wonder what the *solution* is for this system:
>
> ```
>   u  + [v] + {w} = 200.0
>  {u} +  v  + [w] = 190.1
>  [u] + {v} +  w  = 178.8
> ```

This is actually a linear equation system consisting of 6 equations:

```
[u] + [v] = 199
[v] + [w] = 189
[u] + [w] = 178
{u} + {w} = 1.0
{u} + {v} = 1.1
{v} + {w} = 0.8
```

with u = [u] + {u}, ... and so on.
The constant term on the right side has to been adjusted to give positive solutions for the {}s.

The left side of the equation system can be written as matrix for solving in a calculator:

```
[1 1 0 0 0 0]
[0 1 1 0 0 0]
[1 0 1 0 0 0]
[0 0 0 1 0 1]
[0 0 0 1 1 0]
[0 0 0 0 1 1]
```

The solution is [94, 105, 84, 0.65, 0.45, 0.35], which results in:
u = 94.65
v = 105.45
w = 84.35

EMAIL PM FIND QUOTE REPORT

04-02-2020, 10:26 PM (This post was last modified: 04-03-2020 02:56 PM by Bernd Grubert.) **Post: #8**

Bernd Grubert
Member

Posts: 84
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Valentin Albillo Wrote:** ⇨ (04-01-2020 08:52 PM)
>
> **3)** Now, here's another 2nd-degree equation in *X*:
>
> $X^2 + 2 X + 5 I = 0$
>
> with the caveat that this time *X* is not just some scalar value but a square *matrix* and *I* is the corresponding *Identity* matrix. Matricial equations can have any number of roots, including an infinity of them or none at all, and it would be nice to find some *roots* for this equation, if they do exist.

The solutions of the quadratic equation are -1+2*i and -1-2*i. Trivial square matrices that satisfy the equation are diagonal matrices D that contain the solutions as elements in the diagonal, e.g.:

[-1 + 2 i, 0]
[0, -1 - 2 i]

Multiplying those diagonal matrices D from the left with an invertible matrix M of the same dimensions and with its inverse from the right
$X = M D M^{-1}$
is also a solution of the quadratic equation.
This can be seen by multiplying the quadratic equation with M^{-1} from the left and with M from the right:
$M^{-1} X X M + 2 M^{-1} X M + 5 M^{-1} I M =$
$M^{-1} (M D M^{-1}) (M D M^{-1}) M + 2 M^{-1} (M D M^{-1}) M + 5 M^{-1} I M =$
$(M^{-1} M) D (M^{-1} M) D (M^{-1} M) + 2 (M^{-1} M) D (M^{-1} M) + 5 M^{-1} I M =$
$D^2 + 2 D + 5 I$

by making use of the fact that $M^{-1} M = M^{-1} I M = I$.
By definition the result of this expression is 0 since D is a root of the quadratic equation.
Since D is a solution of the equation so is $X = M D M^{-1}$.
Therefore there exist an infinite number of solutions for this equation.

Edit: Added parentheses. Restated the conclusion to make it more clear.

EMAIL PM FIND QUOTE REPORT

04-02-2020, 11:03 PM **Post: #9**

Bernd Grubert
Member

Posts: 84
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Paul Dale Wrote:** (04-02-2020 02:41 AM)
> #7: Distance to planet is 504 light hours. At half impulse this is 1008 hours of travelling time. There are 168 hours in a week. 1008/168 = 6 weeks exactly. Infinite warp speed would be required to get back and save the doomed system.

I agree with Paul. It is not possible to make it back in time.

However Captain Kirk might think (wrongly) that he has enough time due to time dilatation when travelling at half the speed of light (c) to the planet to pick up the ore. Since he is accelerated and decelerated his clock is slower than those in the rest of the universe by a factor of
$1/\text{sqrt}(1 - v^2 / c^2) = 1/\text{sqrt}(1 - 1/4) = 2 / \text{sqrt}(3)$.

For him the travel didn't last 1008 h but only 1008 h * sqrt(3) / 2 ~ 873 h. He might think that he still has ~135 h left to travel back to the solar system and prevent the sun from turning into a supernova. The required travelling speed would be ~ c * 504 / 135 ~ 3.73 c in this case. The warp speed would be ~1.55. (I don't know what happens with time during warp speed - does it reverse?).

On the other hand I don't think Captain Kirk would happen such a blunder :-) and I might also be on a totally wrong track.

EMAIL PM FIND QUOTE REPORT

04-03-2020, 12:55 AM **Post: #10**

Albert Chan
Senior Member

Posts: 874
Joined: Jul 2018

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Valentin Albillo Wrote:** (04-01-2020 08:52 PM)
> **6)** The following expression (where *N > 0* is an *integer* and *log2* is the natural logarithm of 2 = 0.693+):
>
> $\text{Ceil}(2/(2^{1/N} - 1)) = [2*N/\log 2]$
>
> seems to be an *identity* for *all* integer values of n ...

Let gap = $f(N) = 2*N/\log 2 - 2/(2^{1/N} - 1)$
Let w = log(2)/N > 0 → f(N) = g(w) = 2/w - 2/(e^w-1)

Since e^w-1 = w + $w^2/2! + w^3/3!$ + ... > w, we have **f(N) = g(w) > 0**

f(1) ≈ 0.8854
f(2) ≈ 0.9424
f(3) ≈ 0.9615
...
f(2020) ≈ 0.9999

Apply L'Hospital's rule for limit(g(w), w=0):

2*(e^w-1 - w) / (w * (e^w-1))
⇒ 2*(e^w-1) / (w*e^w + (e^w-1))
⇒ 2*e^w / (w*e^w + e^w + e^w)
= 2 / (w + 2)

→ f(∞) = g(0) = 2 / (0+2) = 1

→ **$\text{Ceil}(2/(2^{1/N} - 1)) \ge [2*N/\log 2]$ // LHS > RHS if {2*N/log2} > f(N)**

Example: 37th convergents of log2/2 = 777451915729368 / 2243252046704767

With N = 777451915729368 // note: this may not be the first exception case

LHS = Ceil (2243252046704766.000000000000000106 ...) = 2243252046704767
RHS = Floor(2243252046704766.999999999999999957 ...) = 2243252046704766

EMAIL PM FIND QUOTE REPORT

04-03-2020, 10:31 AM **Post: #11**

EdS2
Member

Posts: 172
Joined: Apr 2014

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Valentin Albillo Wrote:** (04-01-2020 08:52 PM)
>
> - and to top it all, some nice square roots:
>
> ...

Thanks Valentin for a stimulating post. I can't imagine you can find these patterns in decimal expansions of square roots without a brute force search - if that's so, just how many square roots did you search for patterns?

EMAIL PM FIND QUOTE REPORT

04-03-2020, 02:45 PM **Post: #12**

Bernd Grubert
Member

Posts: 84
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

> **Valentin Albillo Wrote:** (04-01-2020 08:52 PM)
>
> **4)** Also, after dealing with finding some roots of the above matricial equation I then considered the converse problem, i.e.: to find an N^{th}-degree equation which has a given *NxN* square matrix as a root. For instance, I wonder what 3^{rd}-degree matricial equation (if any) would have the following 3x3 matrix as a root:
>
> ```
>      2     3     5
>      7    11    13
>     17    19    23
> ```

The matrix X is a root of its own characteristic polynomial:
$X^3 - 36 X^2 - 32 X + 78 I = 0$ where I is the identity matrix.
The reasoning is similar to that is in post #8:
The roots of the characteristic polynomial define the eigenvalues of the matrix X. Therefore a diagonal matrix D which contains the eigenvalues in its diagonal are also roots of the characteristic polynomial. The original matrix X is obtained by multiplication of D with a matrix M made from the eigenvectors from the left and with its inverse M^{-1} from the right:
$X = M D M^{-1}$
Inserting this into the polynomial and using the fact that
$M^{-1} M = M^{-1} I M = I$
as in post #8 shows that X is also a root of the characteristic polynomial.

EMAIL PM FIND QUOTE REPORT

04-10-2020, 01:46 AM (This post was last modified: 04-10-2020 02:18 AM by Valentin Albillo.) **Post: #13**

Valentin Albillo
Senior Member

Posts: 495
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #007 - 2020 April 1st Ramblings

Hi all:

Thanks for your appreciation and your excellent inputs to my 9 ramblings. As stated, I'll give here my original comments on them:

1) *Re solving the system:*

```
   u    + [ v ] + { w } = 200.0
 { u }  +   v   + [ w ] = 190.1
 [ u ]  + { v } +   w   = 178.8
```

Both solutions posted are fully correct, namely **u = 94.65**, **v = 105.45**, **z = 84.35**.

2) *Re solving the equation:*

$x^2 - 10\ [x] + 12.75 = 0$

Again, the posted solution is correct and the equation has indeed four real roots, namely **√29/2**, **√189/2**, **√229/2** and **√269/2**.

It's somewhat funny that simply adding a [] doubles the number of real roots.

3) *Solving the matricial equation:*

$X^2 + 2\ X + 5\ I = 0$

The equivalent scalar equation, $x^2 + 2x + 5 = 0$, has the two conjugate complex roots *(-1, 2)* and *(-1, -2)*. Now, a complex scalar (x,y) can be represented as a 2x2 matrix,

```
| x   -y  |
| y    x  |
```

so, for instance, the complex scalar root (-1, 2) can be represented as

```
| -1  -2  |
|  2  -1  |
```

and this 2x2 matrix is indeed a root of the matricial equation. The same goes for the matrix corresponding to the conjugate scalar root *(-1, -2)* and if desired, this simple *HP-71B* code can be used to check them:

```
1  DESTROY ALL @ OPTION BASE 1 @ DIM A(2,2),B(2,2),C(2,2)
2  MAT INPUT A
3  MAT B=A*A @ MAT B=B+A @ MAT B=B+A @ MAT C=IDN @ MAT C=(5)*C @ MAT B=B+C
4  MAT DISP B;

>RUN
      A(1,1)=?

      -1, -2, 2, -1 [END LINE]

      0 0
      0 0
```

4) *Re finding the matricial equation which has this 3x3 matrix as a root::*

2	3	5
7	11	13
17	19	23

As **Bernd Grubert** posted, by the *Cayley–Hamilton theorem*, a matrix is always a root of its own *characteristic polynomial* (which can be found using the 5-liner program featured in page 6 of my article "HP Article VA012 - HP-71B Math ROM Bakers Dozen (Vol. 2).pdf"), which for this matrix happens to be:

$P(x) = x^3 - 36*x^2 - 32*x + 78$

and indeed the given matrix is a root of this polynomial equation in its matricial form: $\mathbf{P(X) = X^3 - 36*X^2 - 32*X + 78\ I = 0}$.

As an aside, the *eigenvalues* are the roots of the characteristic polynomial, namely *1.10530*, *-1.91703* and *36.81173*. As a check,

- their sum is *36,* which equals the *trace* (sum of the main diagonal elements) of the matrix: 2 + 11 + 23 = *36*
- their product is *-78*, which equals the *determinant* of the matrix, which indeed is *-78*.

5) *Finding some simple way to express the N^{th} Fibonacci number using them, perhaps as simple as a few lines of RPN/RPL code or a short single-line user-defined function, used like this: FNF(1) = 1, FNF(2) = 1, FNF(3) = 2, ..., FNF(10) = 55, ...*

Regrettably no one posted anything about this one so, as usual, I won't comment anything either.

6) *Finding exceptions to the following near-identity (where N > 0 is an integer and log2 is the natural logarithm of 2 = 0.693+):*

```
Ceil(2/(2^(1/N) - 1)) = [2*N/log2]
```

As it happens, the following one is a true *identity*:

```
[(1/(exp(sqr(2)/n)-1)] = [n/sqr(2)-1/2]
```

but the one given above is only a *near-identity* with infinite *exceptions*, the first being **n = 777451915729368**, as we'll see below.

The near-identity may fail only for those *n* which are denominators of the *num/dem* convergents to *2/log(2)* so it suffices to generate said convergents and check their denominators. The following generic *UBASIC* code lists the convergent's numerators *num*, denominators *den* and resulting values *num/den* for any given positive real number *x*, with *max. err* = 10^{-40} and *num, den* being limited to a maximum of 20 digits (all these limits can be trivially modified in the code below):

```
10 point 8 : input X : W=10^(-40) : Md=10^20 : gosub 40
20 print : print N,D : print N/D : print Z : end
30 '
40 V=1 : N=1 : D=0 : Y=10^99 : Z=X
50 C=int(X) : Fp=X-C : if Fp=0 then N=N*C+U : D=D*C+V : return
60 X=1/Fp : S=N : T=D : N=N*C+U : U=S : D=D*C+V : V=T : R=N/D : print N; D, R
70 if abs(R/Z-1)<=W then return
80 if R=Y or max(N,D)>Md then N=U : D=V : return else Y=R : goto 50

run

? 2/log(2)

 2 1    2.0
 3 1    3.0
 23 8    2.875
 26 9    2.8888888888888888888888888888888888888
 75 26    2.8846153846153846153846153846153846153 8
 101 35    2.8857142857142857142857142857142857142 8
 176 61    2.8852459016393442622950819672131147540 9
 277 96    2.8854166666666666666666666666666666666 6
 1007 349    2.8853868194842406876790830945558739255
 2291 794    2.8853904282115869017632241813602015113 3
 10171 3525    2.8853900709219858156028368794326241134 7
 73488 25469    2.8853900820605441909772664808198201735 4
 377611 130870    2.8853900817605257125391609994651180560 8
 1206321 418079    2.8853900817788025708059959959720531287 1
 7615537 2639344    2.8853900817778963257536721245885341205 9
 31668469 10975455    2.8853900817779308466027148760575301889 5
 39284006 13614799    2.8853900817779241544439987692803984840 3
 70952475 24590254    2.8853900817779271413788568430403362242 6
 323093906 111975815    2.8853900817779267782065261145900121378 8
 394046381 136566069    2.8853900817779268435997817290911404940 5
 717140287 248541884    2.8853900817779268141380951308794295612 5
 ...
```

The following 4-line particularized code finds and outputs *the first three exceptions* almost instantly:

```
list

1 point 11 : X=2/log(2) : Z=X : V=1 : N=1 : D=0 : repeat : C=int(X) : F=X-C
2 if F=0 then end else X=1/F : S=N : T=D : N=N*C+U : U=S : D=D*C+V : V=T
3 if ceil(2/(2^(1/D)-1))<>floor(Z*D) then print D
4 until D>10^19

run
```

```
        777451915729368
        140894092055857794
        1526223088619171207
        OK
```

Let's check them:

```
point 10
n=777451915729368:?ceil(2/(2^(1/n)-1)),floor(2*n/log(2))

        224325204670467        224325204670466

n=140894092055857794:?ceil(2/(2^(1/n)-1)),floor(2*n/log(2))

        406534415799078269        406534415799078268

n=1526223088619171207:?ceil(2/(2^(1/n)-1)),floor(2*n/log(2))

        4403748962482230453        4403748962482230452
```

and indeed all three are exceptions, as the left and right sides of the near-identity differ by **1**.

***7)** Finding the minimum warp factor Kirk should engage in order to meet the deadline and avoid massive loss of life:*

As has been pointed out by **Paul Dale**, the ore planet is 504 lighthours away, i.e. 504/24/7 = 3 lightweeks away, and as the *USS Enterprise* is travelling at half impulse (1/2 lightspeed) it will take her 3/(1/2) = *6 weeks* to arrive there so, as the deadline is precisely 6 weeks, it will have *zero* time to make the return trip to the dangerous star and thus Kirk would need to order an **infinite warp factor** to (just!) meet the deadline, which his starship of course can't achieve.

The conundrum seems hopeless but Cpt. Kirk's has always distinguished himself (among many other things) for *not believing in the unwinnable scenario*, as he always does whatever is needed to get the problem solved. He demonstrated as much early in his career, when solving the training *"Kobayashi Maru"* unwinnable scenario by blatant *cheating*, reprogramming the simulation so that he could win it alright.

Same here. By the time Kirk arrives at the ore there's no time to meet the deadline but there's a canon maneuver Kirk has used several times in both the TV series and the movie *"Star Trek IV: The Voyage Home"*, consisting in *time-traveling to the past* by performing the so-called *"slingshot around the sun"* maneuver, defined by Dr. Leonard *"Bones"* McCoy as *"You pick up enough speed, you're in time warp. If you don't – you're fried."*

In essence, the maneuver consists in going very close around a star at *high warp* speed and its main result (if you don't get fried) is that the starship time-travels (even several centuries in the past/future if needed), so Kirk, which has consistently performed this maneuver a number of times using both the *USS Enterprise* and a Klingon *Bird of Prey*, would simply arrive at the ore planet in 6 weeks, beam the ore aboard, and then order high warp towards and around the ore planet's sun (it has one as it's not a rogue planet, it belongs to a system).

The carefully calculated (by Spock) maneuver would succeed in getting the *Enterprise* a sufficient amount of time in the past, then order whatever achievable warp speed is needed to make it to the potentially-exploding star in time. Thus, once in the past, assuming Kirk orders *maximum warp* (14) the *USS Enterprise* will be back in just $504/14^3 \sim$ 11 minutes.

It's cheating, I know, but it's perfectly *canon* and entirely consistent with Kirk's attitude towards *"unwinnable"* scenarios: he doesn't believe in them.

***8)** On whether the function exp(exp(z)) appears to contradict Picard's Little Theorem by missing two values in the complex plane, namely 0 and 1:*

As **J-F Garnier**'s post makes it perfectly clear, *Picard's LT* is safe because while *exp(something)* can never take the value ***0*** it sure can take the value ***1*** for (complex) *nonzero* arguments.

***9)** As for the final spread of misc ramblings:*

- the equation Gamma(x) = Gamma(y) has trivial solutions x = y but surely nontrivial solutions can be found as well ?

Yes, infinitely many. For instance, both *Gamma(**-1.33980199358**)* and *Gamma(**-1.78118797146**)* evaluate to *3.00000000000*, give or take a few ulps.

- the Gamma function grows faster than the exponential function, and we find the positive value of x for which

their graphics cross for the last time, like this (HP-71B code):

```
>FNROOT(1,10,GAMMA(FVAR)-EXP(FVAR))

        7.46360328378

>X=RES @ GAMMA(X),EXP(X)

        1743.41878559       1743.41878559

>GAMMA(LN(RES))

        1743.41878559      ( = RES)
```

- Gamma(Pi) and Gamma(-Pi/2) are surprisingly close when rounded to 2 decimal places

```
>FIX 2 @ GAMMA(PI);GAMMA(-PI/2) @ STD

        2.29       2.30
```

but slightly changing Pi to Pi+$(2/5^2)^2$ above makes both expressions agree to no less than 8 places when truncated.

```
>X=PI+(2/5^2)^2 @ GAMMA(X);GAMMA(-X/2)

        2.30241003575       2.30241009575
```

- the equation `sin(x)+(7*x+1)*cos(x+1) = x` *has a surprising root in the interval [12, 14]*, namely:

```
>FNROOT(12,14,SIN(FVAR)+(7*FVAR+1)*COS(FVAR+1)-FVAR)

         12.9999999016
```

which is a *near-integer* very close to **13**. Indeed, **13** almost satisfies the equation:

```
>SIN(13)+(7*13+1)*COS(13+1)

        12.9999911119      (i.e.: ~13)
```

As for the nice square roots and to answer **EdS2** question, yes, I did use brute-force to find such patterns (and many more !) but I didn't keep a count of how many square roots did I have to search, surely quite a lot. However, the search is quite fast so finding patterns up to, say, **10** digits while looking at (say) **100**-decimal square roots didn't take long and I could indulge in it without wasting undue amounts of time,

That's all, Have a nice weekend and take care !
V.

Find All My HP-related Materials here: **Valentin Albillo's HP Collection**

PM WWW FIND EDIT QUOTE REPORT

04-10-2020, 10:51 AM (This post was last modified: 04-10-2020 10:52 AM by Paul Dale.) **Post: #14**

Paul Dale
Senior Member

Posts: 1,582
Joined: Dec 2013

RE: [VA] SRC #007 - 2020 April 1st Ramblings

For question 5, I came across this paper by Trzaska which explains all.

The transformation from *Φ* to *Ψ* eluded me, unsurprisingly.

Pauli

PM FIND QUOTE REPORT

04-12-2020, 05:13 PM **Post: #15**

Posts: 28

Juan14
Junior Member

Joined: Jan 2014

RE: [VA] SRC #007 - 2020 April 1st Ramblings

I am late but I wanted to post this for question number 5.
If Fn is the n Fibonacci number and φ = (1+√(5))/2 (golden ratio) we have:

Fn = 1/√(5) [φ^n - (-φ)^(-1)] = 1/√(5) [φ^n - (-1)^(-n)*φ^(-n)]
= 1/√(5) [exp(n*ln(φ)) - (-1)^(n)*exp(-n*ln(φ))]
= 2/√(5) [exp(n*ln(φ)) - (-1)^(n)*exp(-n*ln(φ))]/2

When n is even:

Fn = 2/√(5) [exp(n*ln(φ)) - exp(-n*ln(φ))]/2
= 2/√(5) sinh(n*ln(φ))

When n is odd:

Fn = 2/√(5) [exp(n*ln(φ)) + exp(-n*ln(φ))]/2
= 2/√(5) cosh(n*ln(φ))

A program for the 50g.

```
«
5. √ 2 / DUP2 .5 + * LN ROT 2 MOD { COSH } { SINH } IFTE SWAP / 0 RND
»
```

EMAIL PM FIND QUOTE REPORT

Today, 03:01 PM (This post was last modified: Today 04:48 PM by Albert Chan.) **Post: #16**

Albert Chan
Senior Member

Posts: 874
Joined: Jul 2018

RE: [VA] SRC #007 - 2020 April 1st Ramblings

Question 5, for non-negative integer n:

F(n) = floor(φ^n * √(0.2) + 0.5) = floor(cosh(n*ln(φ)) * √(0.8) + 0.1) = floor(sinh(n*ln(φ)) * √(0.8) + 0.9)

EMAIL PM FIND QUOTE REPORT

Today, 06:41 PM **Post: #17**

Valentin Albillo
Senior Member

Posts: 495
Joined: Feb 2015
Warning Level: 0%

RE: [VA] SRC #007 - 2020 April 1st Ramblings

Hi, all:

I see that **Paul Dale** and **Juan14** recently addressed my *Rambling 5*, which was left unanswered so far, so thanks to both of you for addressing it and consequently I'm giving here my original solution and comments (all code is for the *HP-71B + Math ROM*):

> **Quote:**
> 5) Finding some simple way to express the Nth Fibonacci number using them, perhaps as simple as a few lines of RPN/RPL code or a short single-line user-defined function, used like this: FNF(1) = 1, FNF(2) = 1, FNF(3) = 2, ..., FNF(10) = 55, ...

My original solution consists of the following code, which simply initializes two numeric variables (one *real*, one *complex*) to hold some constants for efficiency's sake, and then implements a single-line user-defined function **`FNF(N)`**, which takes `N` (whether integer or not) and uses a single instance of the **Hyperbolic Sine** (`SINH`) function (as required) of a complex argument to directly return the N-th *Fibonacci number*, F_N :

```
DESTROY ALL @ COMPLEX Z @ X=2/SQR(5) @ Z=(LN((1+SQR(5))/2),PI/2)
DEF FNF(N)=REPT(X*SINH(N*Z)/(0,1)^N)
```

Unlike **Juan14**'s solution, mine doesn't distinguish between *odd* `N` and *even* `N`, a single `SINH` suffices for every real `N`. Also notice that *(0,1)* is the value ***i*** (the imaginary unit) and *(1+√5)/2* is φ, the *Golden Ratio*. Let's check it:

`>RUN` *(just once, to initialize the two needed constants)* and now we'll use FNF directly from the command line to compute and display the first 16 *Fibonacci* numbers F_1 .. F_{16}

```
>FOR N=1 TO 16 @ DISP USING "2D,2X,3D.8D";N;FNF(N) @ NEXT N

       1        1.00000000
       2        1.00000000
       3        2.00000000
       4        3.00000000
       5        5.00000000
       6        8.00000000
       7       13.00000000
                ...
      15      610.00000000
      16      987.00000000
```

which are fully correct. The argument N isn't constrained to be an integer so we can compute $F_{20.20}$ like this:

```
>FNF(20.20)

       7448.4401
```

and we can check the basic recurrence $F_N + F_{N+1} = F_{N+2}$ for N = 20.20 like this:

```
>FNF(20.20)+FNF(21.20),FNF(22.20)

       19500.2694   19500.2694
```

Just for fun, let's compute F_{Pi} and F_e :

```
>STD @ FNF(PI),FNF(EXP(1))

       (FPi =) 2.11702705791    (Fe =) 1.7308152698
```

(notice that when rounded to 4 digits we coincidentally have F_e = 1.731 while √3 = 1.732 and thus we have $F_e \sim \sqrt{3}$ *)*

Last, and also for fun, let's compute N such that F_N is a given value, i.e: *inverse Fibonacci*. Add this 3rd program line:

```
3  INPUT F @ FNROOT(1,100,FNF(FVAR)-F) @ GOTO 3
```

and let's try some values:

```
>FIX 8
>RUN
     ? 55
               10.00000000     (F10 = 55)
     ? 6765
               20.00000000     (F20 = 6765)
     ? PI
                4.09041655     (F4.09041655 = Pi)
     ? 2020
               17.48828958     (F17.48828958 = 2020)
```

Regards.
V.

Find All My HP-related Materials here: **Valentin Albillo's HP Collection**