[VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math" Threaded Mode | Linear Mode

02-14-2021, 09:58 PM Post: #1

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

[VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math"

Hi, all ! Happy *San Valentín 2021* to all of you !

Almost a year has elapsed since my previous ***S&SMC***, exactly ***6** years* **6** since I registered in this new-style forum, and today it's ***San Valentin's Day*** so here you are, a brand-new ***Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math"***, once again intended to give you a chance to dust off your chosen HP calculator and actually *use it*, while also flexing your HP-programming muscle *and* your HP-assisted *sleuthing* abilities (*NOT* your *Google Search* proficiency).

Try all ***6 Concoctions 6*** below and see what you can do about them !

Rules:

- Any **HP calc** of your choice may be used (physical or virtual) but I suggest a *Minimum Recommended Model* (**MRM**) for each *Concoction*, which is the simplest model I deem capable of solving it more or less comfortably. Lesser models might, but not comfortably.

- Using anything other than a **physical** or **virtual HP calculator** is strictly disallowed, thus **no** *Wolfram Alpha/Mathematica/Matlab/Pari*, *VBA*, *Excel*. *Pascal*, *C/C#/C++*, *Java*, *Python*, *Lua*, etc., please go elsewhere for that. The idea here is that you actually *use* your HP calc, not your *Lua* on a laptop, so write your code in a language originally supported on some HP calc (***RPN***, ***RPL***, ***71B BASIC***, ***Hppl***, etc).

- *Googling* for the solutions is über-lame and by doing so you're admitting to yourself than you just can't cope and have to cheat to try and save face 🙂.

Concoction the First: Weird limit

*[MRM: **HP-11C** and up]*

Many HP calculators have a function that returns a (pseudo-)random number uniformly distributed between **0** (inclusive) and **1** (exclusive), such as `RND` in the *HP-11C*, *HP-15C*, *HP-71B* and many other models, and `RAN` in the *HP-42S*.

Now get your calc and conduct a test where you generate a couple of random numbers and *add them up*, then if the sum is less than **1** you continue generating and adding up more random numbers one at a time until the sum eventually *exceeds* **1**, while keeping count of just *how many* random numbers did you generate.

For instance, suppose your first test generates *0.87* and *0.65* and their sum is *1.52*, which is *greater* than *1* already so the test is over and **2** random numbers were generated in all. Now you conduct another test and the sequence of random numbers is *0.21*, *0.07*, *0.16*, *0.35*, *0.19* and finally *0.58*, which makes the sum *(1.56)* exceed *1*, so the test is over and we had to generate ***6*** random numbers in all.

The Challenge:

Write a program that simulates the process for a given number of tests and outputs the average count of random numbers generated per test, and then (the *sleuthing* part) use the program to help you answer these questions:

- ***a.*** What do you think is, in the limit, the average count of generated random numbers for their sum to exceed **1** ? Can you recognize what the *exact* value would be ?
- ***b.*** What would the average count be for the sum to exceed **2** ? To exceed **2.021** ? To exceed **Pi** ?

Surprising, isn't it ? But there's more surprises: now get a suitably *fast* HP model (physical or virtual) and conduct a sizable number of tests (say *~100,000*) to find the average count for sums exceeding **3**, **4**, **5** and **5+1/6** (*no* kidding, try it) on the one hand, and for sums exceeding **10**, **15** and **20** on the other, and analyze the results you get, in order to answer these additional questions:

- ***c.*** What do you think is the *asymptotic* expression for the average count needed to exceed *large* values of the sum ?
- ***d.*** Can you explain the *constant* component of said expression ?

Use a *seed* of **1** for the random number generator at the very start of your program (for instance, `RANDOMIZE 1` on the *HP-71B* and `1, SEED` on the *HP-42S*) and give your answers accurate to at least *2-3* digits. Please do *not* use/post any *theoretical* formulas to get the results for now, do it *empirically* by just generating and using actual random numbers.

I'll give my original solutions for both the *HP-42S* and the *HP-71B*, as well as my comments on the results.

Concoction the Second: Weird Sum

*[MRM: **HP-11C** and up]*

Consider the following *infinite **Albillo** sum*:

$$Sum = \sum_{k=1}^{\infty} \frac{2.3.5.7.11.13\ldots.\ p_{k-1}}{(2021+2)(2021+3)(2021+5)\ldots(2021+p_k)}$$

where, other than ***2021***, the coefficients are the ***prime numbers*** p_k in order: p_1 = **2**, p_2 = **3**, p_3 = **5**, p_4 = **7**, ...

Note: Observe that for *k=1* in the sum above, the product in the numerator is the ***empty*** product, thus equal to **1** by definition.

The Challenge:

Write a program to compute and output the sum as accurately as possible, and then (the *sleuthing* part) use your HP calc (perhaps conduct some experiments) to try and attempt to answer this question: ***What's so weird about this sum ?*** *(BTW, forget about Googling for it because I concocted it myself and it's nowhere else to be found AFAIK.)*

I'll give my original solution for the *HP-71B*, as well as my comments on the result.

Concoction the Third: Weird Integral

*[MRM: **HP-15C** and up]*

Consider the following *definite **Albillo** integral*:

$$\int_1^{\varphi} \frac{\Gamma\, ln(\varphi^2 - x)}{\Gamma\, ln\, x + \Gamma\, ln(\varphi^2 - x)}\, dx$$

where ***Γ*** is the ***Gamma*** function, ***ln*** is the *natural logarithm* (i.e., base **e**) and ***φ*** is the ***Golden Ratio*** = *(1+ √5)/2*.

The Challenge:

Use your HP calc to compute (either manually or writing a program to do it) and output the value of the definite integral as accurately as possible, and then (the *sleuthing* part) use your HP calc (perhaps conduct some experiments) to try and attempt to answer this nagging question: ***What's so weird about this integral ?***. *(Again, forget about Googling for it because I concocted it myself and it's nowhere else to be found either.)*

I'll give my original solution for the *HP-71B*, as well as my comments on the result.

Concoction the Fourth: Weird Graph

*[MRM: **HP-PRIME** and other graphing models]*

Consider the following polynomial in two real variables ***x***, ***y***:

$\mathbf{P(x, y)} = 9x^8 + 9y^8 + 36x^2y^6 + 54x^4y^4 + 36x^6y^2 - 100x^6 - 4y^6 - 108x^2y^4 - 204x^4y^2 + 182x^4 - 10y^4 - 84x^2y^2 - 100x^2 - 4y^2 + 9$

The Challenge:

Use your graphing calc (remember: *no Wolfram Alpha*, etc), either by writing and running some program or manually (but then succintly specify the operations performed) to accurately *plot* the resulting *2D* ***graph*** for ***P(x, y) = 0***, and somewhat describe what you see in the graph you get, giving also relevant parameters (say ranges for *x* and *y*, or maybe things like *centers* or *focii* or *radii* or *asymptotes* or *intersections* or *zeros/poles*, whatever.

If you think it might help, you may also attempt to *factorize* the polynomial, but in any case the main question is: ***What's so weird about this graph ?***

I wont post code or manual operations as I don't own any graphing calculators but I'll give the resultant *graphic* you should get, as well as extensive *commentary*.

Concoction the Fifth: Weird Primes

*[MRM: any **fast** physical or virtual HP calc]*

In Milos Forman's 1984 *"Amadeus"* film (winner of 8 *Academy Awards* {aka *Oscars*}, including *Best Picture*) Salieri comments on the perfection of Mozart's music:

> *"**Displace one note and there would be diminishment**, displace one phrase and the structure would fall."*

Now let's bring that observation to the realm of *prime numbers* and consider a prime number so *'Perfectly Prime'* (a *PP* for short, pronounced *"Pepe"*) that changing any single digit would diminish its primeness by turning it into a *composite* number. **Note**: We're talking about *base-10* digits here.

The Challenge:

Write a program *(the faster & shorter, the better)* for your *HP* calc to compute: *(a)* the 5 smallest *PP*, *(b)* the first *PP* greater than 500 million, *(c)* the first *PP* greater than 777,777,777 and only for very *fast* programs/devices, the *second* *PP* greater than 666,666,666.

I'll give my original solution for the *HP-71B*, as well as my comments on the results.

Concoction the Sixth: Weird Year

*[MRM: **HP-11C** and up]*

***Note**: All that follows is stated utterly tongue-in-cheek, not to be taken seriously at all. No offence whatsoever is meant to anyone who's been negatively affected during 2020.*

Unless you've been hiding under a rock last year, surely you're sorely aware that **2020** was a *catastrophic* year and many of you might wonder why did it came that way. I know I did, and being fully convinced that this Universe of ours is a *mathematical* entity subject to mathematical *rules*, I have been analyzing the matter exhaustively using my trusty *HP* calculators and have finally succeeded in unraveling the mystery !! At long last, I now know the *reason* why **2020** was a catastrophic year and of course the reason is of a *mathematical* nature, as expected.

To wit, the reason is that **2020** shares a very *striking numeric property* with many other catastrophic years such as, e.g.: the year **662** (the *Damghan* earthquake killed 40,000 people), the year **458** (the *Antioch* earthquake killed 80,000), the year **1348** (the *Black Death* plague, which killed up to 200 million, was at its apogee), the year **1556** (the *Shaanxi* earthquake killed 830,000) or the year **1849** (the *Great Irish Famine* killed ~1,500,000), to name but a few.

That *can't* be a mere coincidence ! Moreover and just in case this wasn't evidence enough, the number **666** (the infamous *Number of the Beast* of apocalyptic fame) *also* shares that very property as well.

The Challenge:

Use your trusty HP calculator to assist *you* in your sleuthing to try and discover what simple but highly remarkable (*striking*, in fact) numeric *property* all the aforementioned numbers have in common, and then write a program to find out and output a listing of all years between *AD* ***4*** and *AD* ***5000*** (both included) which have this property *(hint: less than 100)*. Of course the listing should include all mentioned past years as well as *future* years thus predicted to be *potentially catastrophic*, up to that limit.

The questions are: *(a)* How many years will be listed in the output ? *(b)* What will be the next predicted potentially catastrophic year after **2020** ?, and *(c)* Should we be concerned ?

As an additional hint to help finding the remarkable shared property, remember *Occam's Razor*: the property can be *unambiguously* stated by saying that the year's number is a *"(five words)"*.

I'll give my solution, two short programs (6 and 7 lines resp.) for the *HP-71B* which produce the listing and also accept a given year in range and demonstrate* whether it has the required numeric property (thus, if it indeed was/might be catastrophic) or not.

- * E.g.: For property *"The year's number is a factorial"* and year *720* you would output *"720 = 1x2x3x4x5x6"* demonstrating the property, while for year *721* you would ouput *"721 = not a factorial"*.

And last but certainly not least, a couple' important ***caveats***:

- Please **do NOT include *CODE* panels** in your replies to this thread, as it makes it difficult for me to generate the online PDF document which will include the whole thread. I expect you'll kindly comply with this requirement but otherwise you'll risk your carefully crafted code appearing truncated or not at all in the final PDF and thus being irretrievably lost from the online document and making your posting it moot. With no CODE panels, to prevent solutions posted before yours spoiling your fun, I advise to simply *avoid* reading any of them before posting your own. Thank you.

- Designing, testing and formatting these *Challenges* and their solutions takes and *awful* lot of time and effort. Hence, if you *do* enjoy them and would like to see more posted in the future, consider participating or commenting on them so that I get *feedback* of your appreciation.

I'll post my ***original solutions*** in a week or so, for you to have enough time (and no excuses) to try them all. That is, if you've got what it takes ...

That's all. **Enjoy !** ... and that's an order !

V.

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

PM WWW FIND EDIT QUOTE REPORT

02-15-2021, 06:43 PM (This post was last modified: 02-15-2021 07:37 PM by Vincent Weber.) **Post: #2**

Vincent Weber
Member

Posts: 175
Joined: May 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Hi Valentin,

Is the Python syntax available on the HP Prime CAS allowed ?

I kind of fell in love with Python, now that it is the mandatory language on educational calculators in France (starting in 2017 with the awesome Numworks, then Casio, then TI, with HP supporting its syntax along with HPPL on the Prime). I am teaching my kids programming with both Scratch and Python, and I have to admit that I like it a lot - Scratch gives the youngest good ideas of structural programming with friendly drag and drop graphical blocks, and Python is quite easy to catch up for older kids.

I find Python more elegant and expressive than any Basic, even the massively enhanced 71B one (but granted, the 71B can interact seamlessly with advanced mathematical features such as matrices. To compete that, Python calculators should provide Numpy, which is not the case yet).

On the Numworks (did I say how much I love this tiny, stylish, fast, well designed machine with awesome software ? Valentin you should love it, as you love beautiful objects) the Python menus allow you to generate skeletons of code with utmost ease.

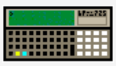
Anyway... I already got the (surprising indeed !) answer for #1. If you ban Python altogether, I shall try with RPN on Free42

Cheers,

Vincent

EMAIL PM FIND QUOTE REPORT

02-15-2021, 08:14 PM **Post: #3**



J-F Garnier
Senior Member

Posts: 484
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

I don't know the reason but I'm particularly attracted by the integral problems (like this previous challenge).
So I tested the "weird" integral:

> **Valentin Albillo Wrote:** (02-14-2021 09:58 PM)
>
> $$\int_1^{\varphi} \frac{\Gamma\, ln(\varphi^2 - x)}{\Gamma\, ln\, x + \Gamma\, ln(\varphi^2 - x)}\, dx$$
>
> where ***Γ*** is the ***Gamma*** function, ***ln*** is the *natural logarithm* (i.e., base ***e***) and ***φ*** is the ***Golden Ratio*** = *(1+ √5)/2*.

The expression to integrate looks complicate, although without any trap, it is not defined at the integral boundaries but that's not a problem with the Romberg algorithm.
I took my old trusted HP-32S and entered the program:

```
LBL I
RCL X
LN
1
-
x!
RCL F
x²
RCL X
-
LN
1
-
x!
+
LASTx
x<>y
/
RTN
```

Then:
`FIX 11`
put the golden ratio in F: `5 SQRT 1 + 2 / STO F`
`FN= I`
`1`
`RCL F`
`∫FN dX`
and quickly got the answer up to at least 11 places.

So what's special? It was fast and easy, without any problem.
Oh wait, it was too easy, too fast. That's *weird*.

[...Investigating a bit...]

Now that I think I found what is "weird", I can even do the calculation by hand, and get the symbolic result without having to identify the numeric result:
[... hidden for now ...]

J-F

robve
Member
Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Challenge one

Note: code indent replaced by U+00A0 (NO-BREAK SPACE) by hand and may not be copyable/usable.

```
HP PRIME
--------
EXPORT MC()
```

```
BEGIN
  LOCAL SUM, N, TRIALS;
  RANDSEED(1);
  N := 0;
  FOR TRIALS FROM 1 TO 100000 DO
    SUM := 0;
    WHILE SUM <= 1 DO
      SUM := SUM + RANDOM();
      N := N + 1;
    END;
  END;
  MSGBOX(N/100000);
END;



BASIC
-----

randomize 1
n=0
for t=1 to 100000
 sum=0
 while sum<=1
  sum=sum+rnd(0)
  n=n+1
 wend
next t
print n/100000



Python
------

import random

def MC():
  random.seed(1)
  n = 0
  for t in range(100000):
    sum = 0
    while sum <= 1:
      sum = sum + random.random()
      n += 1
  print(n/100000)

if __name__ == "__main__":
  MC()
```

a. What do you think is, in the limit, the average count of generated random numbers for their sum to exceed 1 ? Can you recognize what the exact value would be?

2.71959 when trials approaches infinity the exact value is e

b. What would the average count be for the sum to exceed 2 ? To exceed 2.021 ? To exceed Pi?

2 -> 4.67827
2.021 -> 4.71806
3 -> 6.66808
pi -> 6.95027
4 -> 8.66601
5 -> 10.66641
5+1/6 -> 10.99947
10 -> 20.65914
15 -> 30.66700
20 -> 40.66927

c. What do you think is the asymptotic expression for the average count needed to exceed large values of the sum?

I have an idea what kind of inequality in probability theory applies here, which uses EXP() but cannot recall from the top of my head without looking it up. The value appears to be converging to a value between 2k+1/6 and 2(k-1)+e.

d. Can you explain the constant component of said expression?

Working on this...

Challenge two

```
HP PRIME
--------

EXPORT WS2021()
BEGIN
  LOCAL I,J,F,P,SUM,PROD;
  L0 := [2,3,5,7,11,13,17,19];
  FOR P FROM 23 TO 99999 STEP 2 DO
    F := 0;
    FOR I FROM 1 TO SIZE(L0) DO
      IF P MOD L0[ I] = 0 THEN
        F := 1;
        BREAK;
      END;
    END;
    IF F = 0 THEN
      L0 := append(L0, P);
    END;
  END;
  SUM := 0;
  PROD := 1;
  FOR I FROM 1 TO SIZE(L0) DO
    PROD := PROD*L0[ I]/(2021+L0[ I]);
    SUM := SUM + PROD;
  END;
  MSGBOX(SUM);
END;
```

This program displays the value 9.90099737136E-4 for primes up to 99999 (and also for smaller bounds than 99999, such as 97 which is weird).

EDIT
Oops, double check: I read the sum wrong, for the term's numerator has k-1 while the denominator has k. Here is the simple correction. This gives 4.94804552201E-4 and does not change for primes >11 (this sum should converge quickly):

```
EXPORT WS2021()
BEGIN
  LOCAL I,J,F,P,SUM,PROD;
  L0 := [2,3];
  FOR P FROM 5 TO 17 STEP 2 DO
    F := 0;
    FOR I FROM 1 TO SIZE(L0) DO
      IF P MOD L0[ I] = 0 THEN
        F := 1;
        BREAK;
      END;
    END;
    IF F = 0 THEN
      L0 := append(L0, P);
    END;
  END;
  SUM := 0;
  PROD := 1;
  FOR I FROM 1 TO SIZE(L0) DO
    SUM := SUM + PROD/(2021+L0[ I]);
    PROD := PROD*L0[ I]/(2021+L0[ I]);
  END;
  MSGBOX(SUM);
END;
```

Interesting to observe is that the reciprocal of the sum 1 / 4.94804552201E-4 = 2021.

/EDIT

Challenge three

Evaluated on the HP PRIME displays 0.309016994375 (updated to correct a typo in the input).

I don't have the HP-71B (love it), but could use my other BASIC pocket computers since I wrote my own Romberg integration program and Gamma Lanczos approximation in BASIC some time ago. But alas, non-HP machines are not allowed.

EDIT: anyway, let me add this SHARP PC-1350 BASIC program (Valentin, your favorite sharp calc) that produces the answer 0.3090169944 which is perfect to the full 10 digits (updated to correct a typo):

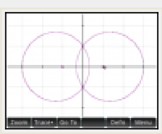
```
' ROMBERG QUADRATURE WITH HIGH PRECISION - Numerical recipes qromb + polint p.134
' see also https://en.wikipedia.org/wiki/Romberg's_...ementation

' Functions to integrate are defined with label "F1", "F2", etc.

' VARIABLES
' A,B range
' F$,F function label (or number) to integrate
' Y result
' E relative error: integral = Y with precision E (attempts E = 1E-10)
' H step size
' N max number of Romberg steps (=10)
' I iteration step
' U current row
' O previous row
' J,S,X scratch
' A(27..26+2*N) scratch auto-array

100 INPUT "a=";A
110 INPUT "b=";B
120 INPUT "F";F
130 E=1E-9,N=10,F$="F"+STR$ F,X=A: GOSUB F$: S=Y,X=B: GOSUB F$
140 H=B-A,O=27,U=O+N,A(O)=H*(S+Y)/2,I=1
150 H=H/2,S=0
160 FOR J=2 TO 2^I STEP 2: X=A+J*H-H: GOSUB F$: S=S+Y: NEXT J
170 A(U)=H*S+A(O)/2,S=4
180 FOR J=1 TO I: A(U+J)=(S*A(U+J-1)-A(O+J-1))/(S-1),S=4*S: NEXT J
190 IF I>1 IF ABS(A(U+I)-A(O+I-1))<E*A(O+I-1) LET Y=A(U+I): PRINT Y: END
200 S=O,O=U,U=S,I=I+1: IF I<N GOTO 150
210 Y=A(O+N-1),S=U+N-2,E=ABS((Y-A(S))/A(S)): PRINT Y,E: END

300 "F1" V=X,X=1.618033989,X=LN(X*X-V): GOSUB "GAMMA": W=Y
310 X=LN V: GOSUB "GAMMA": Z=Y,X=1.618033989,X=LN(X*X-V): GOSUB "GAMMA"
320 Y=W/(Z+Y): RETURN

' gamma(X) -> Y using Lanczos approximation
400 "GAMMA" IF X<=0 LET Y=9E99: RETURN
410 Y=1+76.18009173/(X+1)-86.50532033/(X+2)+24.01409824/(X+3)
420 Y=Y-1.231739572/(X+4)+1.208650974E-3/(X+5)-5.395239385E-6/(X+6)
430 Y=EXP(LN(Y*2.506628275/X)+(X+.5)*LN(X+5.5)-X-5.5): RETURN
```

/EDIT

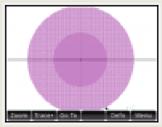
Challenge four

Plot with HP PRIME shows two (cross) eyes. Very funny!

I used the advanced graphing app to enter the X-Y equation. The "eyes" radii are close to the root of 17/6 (updated) and centered at (±4/3,0).



The two "pupils" centered at (±1,0) take some time to converge. They may never computationally converge exactly to a single point.



Now on to the last two challenges. But some information is missing about the PP definition: do you mean that changing any digit to any other digit ALWAYS produces a composite number? So for example 17 is not a PP because 7->9 gives 19?

EDIT

Challenge five

PP #1: 294001
PP #2: 505447
PP #3: 584141
PP #4: 604171
PP #5: 971767

I wrote a Sieve of Eratosthenes program with an addition to filter perfect primes.

The big problem is that HPPL does not permit lists lengths exceeding 20K. So unfortunately I had to rewrite the program in C and run it. I wish HPPL has bitvectors or efficient sets!

```
EXPORT PP()
BEGIN
  LOCAL I,J,K,N,M;
  LOCAL F,D,P,Q,R;
  N := 9999;
  // INIT LIST OF 0 (COMPOSITE) OR 1 (PRIME)
  L0 := MAKELIST(odd(I),I,1,N,1)
  // SIEVE PRIMES
  I := 3;
  WHILE 1 DO
    WHILE I < N AND L0[ I] = 0 DO
      I := I+1;
    END;
    IF I = N THEN
      BREAK;
    END;
    FOR J FROM 2*I TO N STEP I DO
      L0[J] := 0;
    END;
    I := I+2;
  END;
  // FILTER PERFECT PRIMES
  P := 3;
  WHILE 1 DO
    WHILE P < N AND L0[P] = 0 DO
      P := P+1;
    END;
    IF P = N THEN
      BREAK;
    END;
    M := FLOOR(LOG(P));
    F := 1;
    FOR J FROM 0 TO M DO
      I := 10^J;
      // Q = P WITH JTH DIGIT SET TO 0
      Q := FLOOR(P/I/10)*I*10+ROUND(FP(P/I)*I);
      // D = JTH DIGIT OF P (0 to 9)
      D := FLOOR(P/I) MOD 10;
      FOR K FROM 0 TO 9 DO
        IF K <> D AND L0[Q+K*I] THEN
          F := 0;
        END;
      END;
    END;
    IF F THEN
```

```
      PRINT("PERFECT "+P);
    END;
    P := P+2;
  END;
END;
```

I spent a lot of time typing this program in on my HP PRIME (and the other ones). This challenge is meant to enter programs on the HP machine!

Note that the program checks for PP by changing digits. This includes changing the leading digit to any digit 0 to 9, including 0. So for example, 199 -> 099 is not PP. It seems odd to me to change the leading digit to 0 to check for PP.

The second HPPL program is less efficient, but permits a lower bound `B` and upper bound `E` for the search:

```
EXPORT WPP()
BEGIN
  LOCAL B,D,E,F,I;
  LOCAL J,K,P,Q;
  // BEGIN SEARCH AT B
  B := 11;
  // END SEARCH AT E
  E := 9999999;
  FOR P FROM B TO E STEP 2 DO
    IF isprime(P) THEN
      M := FLOOR(LOG(P));
      F := 1;
      FOR J FROM 0 TO M DO
        I := 10^J;
        // Q = P WITH JTH DIGIT SET TO 0
        Q := FLOOR(P/I/10)*I*10+ROUND(FP(P/I)*I);
        // D = JTH DIGIT OF P (0 to 9)
        D := FLOOR(P/I) MOD 10;
        FOR K FROM 0 TO 9 DO
          IF K <> D AND isprime(Q+K*I) THEN
            F := 0;
            BREAK;
          END;
        END;
        IF F=0 THEN
          BREAK;
        END;
      END;
      IF F THEN
        PRINT("PERFECT "+P);
      END;
    END;
  END;
END;
```

The first method (sieving) is more efficient. The asymptotic running time is linear in the max perfect prime *p* we're hunting for, but requires memory of size *p*. The asymptotic running time of the second method is roughly square in *p* but requires constant memory. It takes a few minutes to find the first five perfect primes.

My **C**heat program gives an answer in a fraction of a second:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
  long i, j, k, n = 1000000;
  // sieve (0 = composite, 1 = prime)
  char *s = (char*)malloc(n);

  // init sieve, we should keep only odd values
  for (i = 0; i < n; ++i)
    s[i] = (i & 1);

  // seive for primes
```

```
  for (i = 3; i < n; i += 2)
  {
    while (i < n && s[i] == 0)
      ++i;
    for (j = 2*i; j < n; j += i)
      s[j] = 0;
  }

  // sieve for perfect primes
  for (i = 3; i < n; i += 2)
  {
    while (i < n && s[i] == 0)
      ++i;
    if (i < n)
    {
      long m = floor(log10(i));
      long p = 1; // p = 10^j
      char h = 1; // h = 1 if i is perfect
      // for each digit in prime i, from least to most significant
      for (j = 0; j <= m; ++j, p *= 10)
      {
        // q = prime i with jth digit zero
        long q = i%p + i/p/10*p*10;
        // d = jth digit of prime i
        long d = i/p%10;
        // twiddle jth digit and check for prime
        for (k = 0; k <= 9; ++k)
          if (k != d && s[q+k*p])
            h = 0;
      }
      if (h)
        printf("perfect %ld\n", i);
    }
  }
  free(s);
}
```

b. 500004469 (first PP > 500000000)

c. 777781429 (first PP > 777777777)

d. 666999929 (second PP > 666666666)

For small primes with k digits such as k=1, k=2, ..., k=5 digits there are no perfect primes in base 10, because primes with k digits are "too close" (as in a Hamming distance kind of way). The prime number theorem tells us that primes are distributed roughly as N/log(N) so that a randomly picked integer not greater than N has a probability of 1/log(N) to be prime. A perfect prime of k digits can be perturbed by its definition to generate 9k distinct composite integers. Roughly, the chance that an integer of k digits is prime is 1/log(10^k)=0.43/k. The chance that the 9k integers are all composite is (1-0.4343/k)^(9k), assuming k is sufficiently large. It turns out that the chance approaches 2% for large k and is half that for small k (though the log constant is somewhat arbitrary). More importantly, there are also far more integers to pick as potential perfect primes for large k. Based on this, it seems reasonable to see perfect primes for large k and there are infinitely many of them.

/EDIT

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

02-15-2021, 10:20 PM (This post was last modified: 02-15-2021 11:06 PM by J-F Garnier.) **Post: #5**

J-F Garnier
Senior Member

Posts: 484
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **robve Wrote:** (02-15-2021 09:51 PM)
> **Challenge three**
> Evaluated on the HP PRIME displays 5.61670944148E-2.

Can you double check?
I got .30901699438 on my 32S.

And what do you think is *weird*?

J-F

[edited: ok I see your mistake, you missed the LN in the numerator]

EMAIL PM WWW FIND QUOTE REPORT

02-15-2021, 11:08 PM (This post was last modified: 02-16-2021 12:03 AM by ramon_ea1gth.) **Post: #6**

ramon_ea1gth
Junior Member

Posts: 35
Joined: Mar 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

I tried the weird limit in my physical HP 50g in User-RPL, reaching 100,000 iterations. When I saw the outcome going to e, some kind of law of large numbers came to my mind.

Input stack:
2: number of tests, CNT
1: limit to exceed, XCD (e.g., 1, pi)

My code:

```
« 0 → CNT XCD TOT
« 1 RDZ
1 CNT FOR I
RAND 1 → TMP NR
« DO RAND 'TMP' STO+ 1 'NR' STO+
UNTIL TMP XCD >
END
NR 'TOT' STO+
»
NEXT
TOT CNT /
»
»
```

Ramón
Valladolid, Spain
TI-50, Casio fx-180P, HP48GX, HP50g, HP Prime G2

EMAIL PM WWW FIND QUOTE REPORT

02-15-2021, 11:14 PM (This post was last modified: 02-15-2021 11:32 PM by Valentin Albillo.) **Post: #7**

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

.
Thanks for your interest and for participating, **some quick answers to your questions**:

> **Vincent Weber Wrote:**
> Is the Python syntax available on the HP Prime CAS allowed ? Smile [...] If you ban Python altogether, I shall try with RPN on Free42 Smile

If it's available for the *HP Prime* then it's technically allowed but I'd rather have a program in *Hppl* for the *Prime* or in *RPN/RPL/BASIC/FORTH/Assembler/etc.* for any other HP calc. Anyway, your choice.

> **robve Wrote:**
> But some information is missing about the PP definition: do you mean that changing any digit to any other digit ALWAYS produces a composite number? **So for example 17 is not a PP because 7->9 gives 19?**

Yep. If changing any single digit to any other digit results in a prime number, the original one is not a *PP*.

Regards.
V.

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

PM WWW FIND EDIT QUOTE REPORT

02-16-2021, 09:34 AM **Post: #8**

Vincent Weber
Member

Posts: 175
Joined: May 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Thanks Valentin.
Anyway it seems that similar solutions to mine have already been posted. This one is actually fairly easy, from a programming perspective, if not from a theorical perspective - but you didn't allow theory here :)

Cheers,

Vincent

EMAIL PM FIND QUOTE REPORT

02-16-2021, 10:56 AM **Post: #9**

Werner
Senior Member

Posts: 591
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Free42 code for challenges 1,2 and 3:

VA#1: average nr of random numbers in X runs to sum over Y

e.g. 2 ENTER 100000 XEQ"VA1" -> 4.67827

```
00 { 39-Byte Prgm }
01▸LBL "VA1"
02 LSTO "X"
03 LSTO "C"
04 SIGN
05 SEED
06 CLX
07 STO ST Z
08▸LBL 02
09 ISG ST Z
10 X<>Y
11 RAN
12 +
13 X≤Y?
14 GTO 02
15 CLX
16 DSE "C"
17 GTO 02
18 R^
19 RCL÷ "X"
20 END
```

results identical to robve's post. I would say that for 'larger' n, the limit is 2(n+1/3)..

```
00 { 73-Byte Prgm }
01▸LBL "VA2"
02 1
03 LSTO "P" @ term
04 CLX
05 LSTO "S" @ sum
06 2
07 XEQ 14
08 3
09▸LBL 03
10 XEQ "P?"
11 X=0?
12 GTO 00
13 R↓
14 XEQ 14
15 X=Y?
```

```
16 RTN
17 R↓
18▸LBL 00
19 R↓
20 2
21 +
22 GTO 03
23▸LBL 14
24 RCL "P"
25 2021
26 RCL ST Z
27 +
28 ÷
29 ×
30 STO "P"
31 X<> ST L
32 X<> "S"
33 STO+ "S"
34 RCL "S"
35 END
```

The P? primality testing program returns X and 0 when not prime, 1 when prime. You may use eg

```
00 { 114-Byte Prgm }
01>LBL "P?"
02 ENTER
03 STO ST Z
04 2
05 X=Y?
06 GTO 01
07 X>Y?
08 SIGN
09 MOD
10 X=0?
11 RTN
12 CLX
13 3
14 X=Y?
15 GTO 01
16 MOD
17 X=0?
18 RTN
19 CLX
20 5
21 X=Y?
22 GTO 01
23 MOD
24 X=0?
25 RTN
26 CLX
27 7
28>LBL 03
29 X^2
30 X>Y?
31 GTO 01
32 SQRT
33 MOD
34 X=0?
35 RTN
36 CLX
37 4
38 LASTX
39 +
40 MOD
41 X=0?
42 RTN
43 CLX
44 2
45 LASTX
46 +
47 MOD
48 X=0?
```

```
49 RTN
50 CLX
51 4
52 LASTX
53 +
54 MOD
55 X=0?
56 RTN
57 CLX
58 2
59 LASTX
60 +
61 MOD
62 X=0?
63 RTN
64 CLX
65 4
66 LASTX
67 +
68 MOD
69 X=0?
70 RTN
71 CLX
72 6
73 LASTX
74 +
75 MOD
76 X=0?
77 RTN
78 CLX
79 2
80 LASTX
81 +
82 MOD
83 X=0?
84 RTN
85 CLX
86 6
87 LASTX
88 +
89 GTO 03
90>LBL 01
91 SIGN
92 END
```

@robve: you made the same mistake I did at first.. read the formula carefully, the nominator goes to pk-1 and the denominator to pk.

The sum converges quickly to 4.948045522...E-04, which is exactly 1/2021. Why? I'm sure Valentin will enlighten us in a few days ;-)

VA#3: integral
Program to be used with INTEG

```
00 { 37-Byte Prgm }
01▸LBL "VA3"
02 MVAR "X"
03 1.25
04 SQRT
05 1.5
06 +
07 RCL- "X"
08 LN
09 GAMMA
10 RCL "X"
11 LN
12 GAMMA
13 RCL+ ST Y
14 ÷
15 END
```

Making use of the fact that φ^2 = φ+1

Same observations as J-F.. though the fact that it's easy to see what the value is, does not explain (to me) why it converges so quickly.

Cheers, Werner

EMAIL PM FIND QUOTE REPORT

02-16-2021, 03:11 PM **Post: #10**

rprosperi
Senior Member

Posts: 4,559
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Werner Wrote:** (02-16-2021 10:56 AM)
> The sum converges quickly to 4.948045522...E-04, which is exactly 1/2021. Why? I'm sure Valentin will enlighten us in a few days ;-)

The first S&S Math Challenge in 2021?

--Bob Prosperi

EMAIL PM FIND QUOTE REPORT

02-16-2021, 10:12 PM **Post: #11**

Namir
Senior Member

Posts: 758
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Regarding the weird sum that uses primes, I used Matlab to calculate each term and the updated sum. The sum seems to reach the asymptotic value of 9.90099737e-04.

The value of 2021, which I will call the **Albillo Parameter** helps the summation to reach the assumptotic value at the prime of 7. The smaller values of the **Albillo Parameter** will delay the summation from reach an assymptotic value (which depends on the Albillo Parameter). For example if the Albillo Parameter is, say, 101, the summation reaches the assymptotic value of 2.00059191e-02 at prime of 23.

My main focus is on the sumamtion in general and the role that the Albillo Parameter plays.

My own question is what is the relationship between the summation's assymptotic value and the value of the **Albillo Parameter**?

Cool summation!!!!

:-)

Namir

EMAIL PM FIND QUOTE REPORT

02-17-2021, 09:44 PM **Post: #12**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

For challenge five, which I have finished, it does not seem possible to implement efficient sets on the HP PRIME because lists are limited to only 20K elements. There is no bitvector or efficient set type that we can use.

Because of this limitation I rewrote the program in **C** (sorry for **C**heating...). With this I found:

perfect 294001
perfect 505447
perfect 584141
perfect 604171
perfect 971767

Note that the program checks for PP by changing one digit of a prime at a time to check if that number is prime or not. This includes changing the leading digit to any digit 0 to 9, including 0. So for example, 199 -> 099 is not PP. It seems odd to me to change the leading digit to 0 to check for PP.

I updated my post with **EDIT**s to add my answer as well as make a correction to the weird sum (I misread k-1).

- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

02-18-2021, 09:22 AM (This post was last modified: 02-18-2021 09:27 AM by J-F Garnier.) **Post: #13**

J-F Garnier
Senior Member

Posts: 484
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Werner Wrote:** (02-16-2021 10:56 AM)
> VA#3: integral
> [..]
> Making use of the fact that φ^2 = φ+1
> Same observations as J-F.. though the fact that it's easy to see what the value is, does not explain (to me) why it converges so quickly.

I thought the other way around: I understood that it's so fast because each sample set used by the algorithm gives the *exact* value of the integral, so the algorithm terminates after just the first 2 estimations (7 samples used).
Then I thought that even *one* sample is enough to give the exact value, using the midpoint that contributes to 1/2, hence the result.

And yes, φ^2 = φ+1 is the key :-)

J-F

EMAIL PM WWW FIND QUOTE REPORT

02-18-2021, 10:28 AM **Post: #14**

Werner
Senior Member

Posts: 591
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

The key to the value of the integral in #3 is to realize that the integral is symmetric in the integral boundaries. Basically it is

integral(a,b,f(a+b-x)/(f(x) + f(a+b-x)),dx)

if you substitute y = a+b-x, dy=-dx, you get

integral(b,a,-f(y)/((y) + f(a+b-y)),dy) = integral(a,b,f(y)/((y) + f(a+b-y)),dy)

Meaning the value of the integral is the sum of both divided by two, or (b-a)/2, whatever f(x) is. So here it is (φ-1)/2.

Cheers, Werner

EMAIL PM FIND QUOTE REPORT

02-18-2021, 02:41 PM **Post: #15**

Gerson W. Barbosa
Senior Member

Posts: 1,369
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

This should provide all the answers to #6:

```
20

« { } 1 DUP2 ROT 5 ROLL
START NEXTPRIME + LASTARG NIP
NEXT DROP SQ DUP SIZE 2 SWAP
FOR i 1 OVER SIZE 1 + i -
FOR j DUP j DUP i + 1 - SUB ∑LIST ROT + SWAP
NEXT
NEXT + SORT
»

EVAL
```

```
« REVLIST 1 119
START TAIL
NEXT SORT
»

EVAL
```

Perhaps 19 or less would suffice as the argument for the first program, but I have not enough time to check this out right now.

Best regards,

Gerson.

EMAIL PM FIND QUOTE REPORT

02-18-2021, 07:46 PM (This post was last modified: 02-19-2021 04:10 PM by Nihotte(lma).) **Post: #16**

Nihotte(lma)
Member

Posts: 64
Joined: Mar 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (02-14-2021 09:58 PM)
>
> **Hi, all ! Happy *San Valentín 2021* to all of you !**
>
> ...
> That's all. **Enjoy !** ... and that's an order !
>
> **V.**

UPDATED 2021.02.19 - 04.10PM to comply with the mention : Please do NOT include CODE panels in your replies to this thread

Hi,

Just to say, I've taken part of the challenge too !
However, I've progressed more slowly than you all...

For a. and b. points

I began with my HP35s :

** LBL V
INPUT S // 1 for stop
INPUT L // 100000 for loops
CLΣ
1
SEED
RCL L
STO I
009 CLx
STO J
STO C
012 1
STO + J
RANDOM
STO + C
RCL C
RCL S
x > y
GTO V012
0
RCL J
Σ+
DSE I
GTO V009
RCL L
RCL S
x-mean
RTN

XEQ V001
with S=1 and L=1E5 gave 2.71959 about 6 hours later

Then, I've brought out my 48G

```
<<
DEPTH DROPN
0 'SUM' STO
SEED RDZ
1 LOOPS
FOR i
0 DUP
DO
SWAP
1 +
SWAP
RAND +
DUP
TOP
UNTIL >= END
DROP 'SUM' STO+
NEXT
TOP LOOPS SUM OVER /
>>
```

with :
VAW is the program above
1 'TOP' STO // 1, 2 .. 20
1 'SEED' STO
100000 'LOOPS' STO
And the run of VAW gives
1
100000
2.71959
in the stack
and SUM is 271959

The result appears more rapidly than with the HP35s, of course
6x faster, perhaps

Unsurprisingly, my results match those of robve's post for the successive limit values 1, 2 .. 20 retained

------

The result for a limit of 1 seems to be close to e.
But, I've tested something on my HP10BII+
I decided to enter all couple of result by Σ+ in the calculator

C STAT
1 INPUT 2.71959 and Σ+
2 INPUT 4.67827 and Σ+
2.021 INPUT 4.71806 and Σ+
3 INPUT 6.66808 and Σ+
pi INPUT 6.95027 and Σ+
4 INPUT 8.66601 and Σ+
5 INPUT 10.66641 and Σ+
5+1/6 INPUT 10.99947 and Σ+
10 INPUT 20.65914 and Σ+
15 INPUT 30.66700 and Σ+
20 INPUT 40.66927 and Σ+
then,
[BLUE] REGR and [-] to select 0 - bESt Fit [INPUT]
and,
2 [ORANGE] ŷ,m displaying bESt Fit with the choice of 1 - LinEAr and the result of 4.6773856...
followed by [ORANGE] ^x,r resulting in 2 and [SWAP] giving 0,999999... as a correlation coefficient to describe the goodness of the fit.

Now here is where I am going with my initial idea :

1000 [ORANGE] ŷ,m 1999.68225...
100000 [ORANGE] ŷ,m 199900.966545...
200000 [ORANGE] ŷ,m 399801.25371...
1E9 [ORANGE] ŷ,m 1999002872.33...
9999999999 [ORANGE] ŷ,m 19990028715,2...
I don't know what to think but, it's giving a result near of 2*x decreased by something that is proportional to x (based on m, in fact) !!!

EMAIL PM FIND QUOTE REPORT

02-19-2021, 12:23 AM (This post was last modified: 02-19-2021 11:23 PM by Gerson W. Barbosa.) **Post: #17**

Gerson W. Barbosa
Senior Member

Posts: 1,369
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Gerson W. Barbosa Wrote:** (02-18-2021 02:41 PM)
> Perhaps 19 or less would suffice as the argument for the first program...

Indeed 19 is enough:

```
19

« { } 1 DUP2 ROT 5 ROLL
START NEXTPRIME + LASTARG NIP
NEXT DROP SQ DUP SIZE 2 SWAP
FOR i 1 OVER SIZE 1 + i -
FOR j DUP j DUP i + 1 - SUB ∑LIST ROT + SWAP
NEXT
NEXT + SORT
»

EVAL

« REVLIST 1 99
START TAIL
NEXT SORT
»

EVAL

→

{ 4 9 13 25 34 38 49 74 83 87 121 169 170 195 204 208 289 290 339 361 364 373 377 458 529 579 628
650 653 662 666 819 841 890 940 961 989 1014 1023 1027 1179 1348 1369 1370 1469 1518 1543 1552 1556
1681 1731 1802 1849 2020 2189 2209 2310 2330 2331 2359 2384 2393 2397 2692 2809 2981 3050 3150 3171
3271 3320 3345 3354 3358 3481 3530 3700 3721 4011 4058 4061 4350 4489 4519 4640 4689 4714 4723 4727
4852 4899 }
```

P. S.:

2020 is in the list because $2020 = 17^2 + 19^2 + 23^2 + 29^2$. In fact every number in the list is either a squared prime or the sum of an ordered sequence thereof.

The number of consecutive prime squares starting with 4 (2^2) required to safely producing a list of all years that share this property, up to a certain year, can be mathematically determined, but that's way beyond my math skills. The following program uses a borrowed formula.

```
5000

« DUPDUP 4 * SWAP LN SQ / 3 XROOT 3 * CEIL { } 1 DUP2 ROT 5 ROLL
START NEXTPRIME + LASTARG NIP
NEXT DROP SQ DUP SIZE 2 SWAP
FOR i 1 OVER SIZE 1 + i -
FOR j DUP j DUP i + 1 - SUB ∑LIST ROT + SWAP
NEXT
NEXT + SORT REVLIST
WHILE DUP HEAD PICK3 >
```

```
REPEAT TAIL
END REVLIST NIP
»

EVAL

Checksum: # 180Ch
Length: 213.5 bytes
```

This will return a list with 91 elements, same as above.

EMAIL PM FIND QUOTE REPORT

02-19-2021, 09:15 PM **Post: #18**

Nihotte(lma)
Member

Posts: 64
Joined: Mar 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (02-14-2021 09:58 PM)
>
> **Hi, all ! Happy *San Valentín 2021* to all of you !**
>
> ## Concoction the Third: Weird Integral
>
> *[MRM: **HP-15C** and up]*
>
> Consider the following *definite **Albillo** integral*:
>
> $$\int_1^{\varphi} \frac{\Gamma\, ln(\varphi^2 - x)}{\Gamma\, ln\, x + \Gamma\, ln(\varphi^2 - x)}\, dx$$
>
> where ***Γ*** is the ***Gamma*** function, ***ln*** is the *natural logarithm* (i.e., base ***e***) and ***φ*** is the ***Golden Ratio*** = *(1+ √5)/2*.
>
> **The Challenge:**
>
> Use your HP calc to compute (either manually or writing a program to do it) and output the value of the definite integral as accurately as possible, and then (the *sleuthing* part) use your HP calc (perhaps conduct some experiments) to try and attempt to answer this nagging question: ***What's so weird about this integral ?***. *(Again, forget about Googling for it because I concocted it myself and it's nowhere else to be found either.)*
>
> That's all. **Enjoy !** ... and that's an order !
>
> **V.**

Here is the program I used on my HP35s for the Concoction the Third and the Weird Integral:

LBL W
RCL P
x²
RCL X
-
LN
!
ENTER
ENTER // 2 times because ENTER has disabled moves in the stack...
RCL X
LN
!
+
/
RTN

With FN= W
1 and RCL P (where P is 1.61803398875)
∫FN d X
INTEGRATING gives 3.09016994375E-1 and it seems to be 1/(2*φ)

02-20-2021, 12:32 AM Post: #19

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Hi, **all**:

First of all, **thanks** a lot for your interest and your participation in this challenge, really much appreciated. As stated in my OP, I'll post *my original solutions* and extensive comments in a few days but as there's still a number of questions left unanswered so far, I'll give you all a

Last Chance

to address and complete them before I do. This includes the following:

Concoction the First: Weird limit

Point **d** is still unaddressed: ***d.*** *Can you explain the constant component of said [asymptotic] expression ?*

Concoction the Second: Weird Sum

The main question is wholly unaddressed: ***What's so weird about this sum ?***. Perhaps a little sleuthing (i.e.: conducting some experiments) would be of help.

Concoction the Fourth: Weird Graph

This is mostly unaddressed. In particular no program or description of the operations required to produce the graph have been posted so far and worse, no one has produced and posted the ***graph*** *itself*, i.e.: an actual *image* of it. This is the kind of functionality for which *graphics calculators* are intended and I included this part specifically to give *HP Prime*'s or *RPL*-models' users some opportunity to show off their calculator's worthiness for this challenge.

Also, no attempts to *factorize* the polynomial have been reported, either successful or unsuccessful. This is the kind of functionality for which **CAD** is intended. Why don't you give it a try ? *Hint:* It would help to answer the main question.

Finally, as for the main question proper, ***What's so weird about this graph?***, it's still left unanswered. Apart from its so-described *"funny"* appearance, there's more to the graph than it seems at first sight. Some sleuthing would surely help.

Concoction the Sixth: Weird Year

Both *RPL* code and the resulting list of the years have been produced (without comments or explanation), but nearly all the questions have been left unanswered, i.e.:

- What is the *"simple but highly remarkable (striking, in fact) numeric property"* ?
- (a) How many years will be listed in the output ? ,
- (b) What will be the next predicted potentially catastrophic year after 2020 ?,
- (c) Should we be concerned ?

Also, although not explicitly asked, no program has been produced to accept a given year in range and *demonstrate** whether it has the required numeric property (thus, if it indeed was/might be catastrophic) or not, which would be nice as I state that my original solution *does* exactly this.

** E.g.: For property "The year's number is a factorial" and year 720 you would output "720 = 1x2x3x4x5x6" demonstrating the property, while for year 721 you would ouput "721 = not a factorial"*

Finally, programs written in other than *RPL* would be welcome for variety and to let readers better understand what the code does and how their *RPN* calcs (say) would deal with the problem.

As stated, **Last Chance**. Thanks and best regards. :)
V.

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

02-20-2021, 04:45 AM **Post: #20**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

For challenge two (weird sum), 1 / 4.94804552201E-4 = 2021.

Pretty neat!

- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

02-20-2021, 12:29 PM **Post: #21**

J-F Garnier
Senior Member

Posts: 484
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (02-20-2021 12:32 AM)
>
> **Concoction the Second: Weird Sum**
>
> The main question is wholly unaddressed: ***What's so weird about this sum ?***. Perhaps a little sleuthing (i.e.: conducting some experiments) would be of help.

I can't explain how this sum converges to 1/2021.
I tried other constant value than 2021, it still worked.

Then I asked myself: are the prime numbers the key for the convergence?
So I calculated the terms, using the suite of natural integers instead of the primes.

1 x 2 x 3 x 4 x 5 x ... (n-1)

(A+2) x (A+3) x (A+4) x ... (A+n)

Where A is the constant, 2021.

```
10 A=2021 @ S=0 @ X=1
20 FOR I=1 TO 10
30 X=X*I/(A+I+1) ! next term
40 S=S+X ! sum
50 NEXT I
60 DISP S, 1/S
```

And guess what? The sum seems to still converge to 1/A

Quite *weird*, no?

J-F

EMAIL PM WWW FIND QUOTE REPORT

02-20-2021, 02:37 PM (This post was last modified: 02-20-2021 02:57 PM by C.Ret.) **Post: #22**

C.Ret
Junior Member

Posts: 47
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (02-20-2021 12:32 AM)
>
> **Hi, all**:
>
> **Concoction the Fourth: Weird Graph**
>
> This is mostly unaddressed. In particular no program or description of the operations required to produce the graph have been posted so far and worse, no one has produced and posted the ***graph*** *itself*, i.e.: an actual *image* of it. This is the kind of functionality for which *graphics calculators* are intended and I included this part specifically to give *HP Prime*'s or *RPL*-models' users some opportunity to show off their calculator's worthiness for this challenge.
>
> Also, no attempts to *factorize* the polynomial have been reported, either successful or unsuccessful. This is the

kind of functionality for which **CAD** is intended. Why don't you give it a try ? *Hint:* It would help to answer the main question.

Finally, as for the main question proper, ***What's so weird about this graph?***, it's still left unanswered. Apart from its so-described *"funny"* appearance, there's more to the graph than it seems at first sight. Some sleuthing would surely help.

Thanks and best regards.
V.

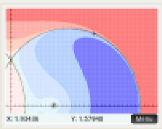
I spent so much time on these wierd graph that my valentine leave me and she is actualy going out with a former best friend of mine.

The good thing is that I am now full time free of wasting plenty of time on my followed and trusty HP Color Graphing Calculator.

Another good fact is that this polynomial is perfectly even for both arguments x and y this simply greatly the investigation:

$$Po(x,y) = Po(-x,y) = Po(x,-y) = Po(-x,-y)$$

One may investigate it out in the first graph quadrant aka x>=0 and y>=0 only since other quadrants are exact symmetric reflections.



I was unable to directly factorize this wierd Polynomial. But by factorizing specific sub-expressions I get information about where to look for roots on specific axis:

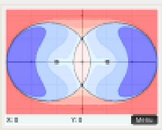
$$Po(x,0) = 9x^8 - 100x^6 + 182x^4 - 100x^2 + 9 = (x-3)(x-1)^2(x+1)^2(x+3)(3x-1)(3x+1)$$

This clearly indicate that roots on the Ox axis are found at abscissas -3 -1 -1/3 +1 +1/3 +3.
But the squared factors trigger me, I was under alert, there is double root there. Nodes points. Are same time hard to trace on graphics !

$$Po(0,y) = 9y^8 - 4y^6 - 10y^4 - 4y^2 + 9 = (y-1)^2(y+1)^2(3y^2-2y+3)(3y^2+2y+3)$$

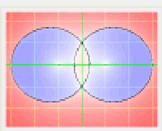
Fortunately, $3y^2?2y+3$ have no real roots , only pure complex zeros. So the only expected zero on the Oy axis are at -1 and +1. Again, double roots alert there .

I was on the way to investigate other specify factorization of $Po(-3,y)$ $Po(-1,y)$ $Po(-\frac{1}{3},y)$ $\cdots$but I realize how powerful is a Graphing Calculator and his Advanced Graphics Application:



The position $Po(x,y) = 0$ are marked by black traces. The graphic windows is from (-3.2,2.4) to (3.2,-2.4). Ticks are 1 unit for both x and y axis. Square zoom 9.
The red color indicate positive values of$Po(x,y)$, the darker the highest.
The blue color indicate negative values of $Po(x,y)$, the darker the deepest.

The Advanced Graphic App makes a decent job.
But I spent a few hours developing my own way of doing it:



The weirdest with this polynomial is to catch the two zeros at (-1,0) and (+1,0) since the polynomial reach zeros there but without any change of sign , all surrounding values stay negative except at the exact zeros points.

Thank a lot for this amazing pulzze. I really spend a good time playing on it !

PM FIND QUOTE REPORT

02-20-2021, 09:47 PM (This post was last modified: 02-20-2021 10:02 PM by robve.) **Post: #23**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

The weird primes concoction (aka challenge five) depends on the decimal number base. One could argue that a "perfect" property definition should not depend on the number base. The choice of base affects the distribution of base-perfect primes. The prime number theorem sheds some light on the frequency of perfect primes in general.

1. We should observe more primes that meet the definition of base-perfectness when smaller number bases are used than larger number bases.

2. For small primes with k digits such as k=1, k=2, ..., k=5 digits there are no perfect primes (in base 10), because primes with k digits are "too close" (as in a Hamming distance kind of way). The prime number theorem tells us that primes are distributed roughly as N/log(N) so that a randomly picked integer not greater than N has a probability of 1/log(N) to be prime. A perfect prime of k digits can be perturbed by its definition to generate 9k distinct composite integers. Roughly, the chance that an integer of k digits is prime is 1/log(10^k)=0.43/k. The chance that the 9k integers are all composite is (1-0.4343/k)^(9k), assuming k is sufficiently large. It turns out that the chance approaches 2% for large k and is half that for small k (though the log constant is somewhat arbitrary). More importantly, there are also far more integers to pick as potential perfect primes for large k. Based on this, it seems reasonable to see perfect primes for large k and there are infinitely many of them.

The first base-perfect primes for base 2 to 19 are (shown is the minimum number of base digits for the perfect prime shown in decimal):

```
base | digits | first PP   |
2    | 7      | 127        |
3    | 3      | 13         |
4    | 5      | 373        |
5    | 3      | 83         |
6    | 6      | 28151      |
7    | 3      | 223        |
8    | 5      | 6211       |
9    | 4      | 2789       |
10   | 6      | 294001     |
11   | 4      | 3347       |
12   | 7      | 20837899   |
13   | 4      | 4751       |
14   | 6      | 6588721    |
15   | 5      | 484439     |
16   | 5      | 862789     |
17   | 4      | 10513      |
18   | 8      | 2078920243 |
19   | 4      | 10909      |
```

A very interesting pattern is emerging that has an explanation.

The C "perp" program (note: indent spacing U+00A0):

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char **argv)
{
  if (argc > 2)
  {
    long i, j, k;
    const long base = atoi(argv[1]); // perfect prime base
    const long maxd = atoi(argv[2]); // up to primes of maxd digits
    const double lnbase = log(base);
    const long max = pow(base, maxd); // primes up to max (exclusive)
    char *sieve = (char*)malloc(max); // sieve (0 = composite, 1 = prime)
```

```
    // init sieve, we should keep only odd values
    for (i = 0; i < max; ++i)
      sieve[i] = (i & 1);

    // seive for primes
    for (i = 3; i < max; i += 2)
    {
      while (i < max && sieve[i] == 0)
        ++i;
      for (j = 2*i; j < max; j += i)
        sieve[j] = 0;
    }

    // sieve for perfect primes
    for (i = 3; i < max; i += 2)
    {
      while (i < max && sieve[i] == 0)
        ++i;
      if (i < max)
      {
        long m = floor(log(i)/lnbase); // m = number of digits of i
        long p = 1; // p = base^j
        char h = 1; // h = 1 if i is perfect
        // for each jth digit in prime i, from least to most significant
        for (j = 0; j <= m; ++j, p *= base)
        {
          // q = prime i with jth digit zero
          long q = i%p + i/p/base*p*base;
          // d = jth digit of prime i
          long d = i/p%base;
          // twiddle jth digit and check for prime
          for (k = 0; k < base; ++k)
            if (k != d && sieve[q+k*p])
              h = 0;
        }
        if (h)
          printf("perfect_%ld %ld\n", base, i);
      }
    }

    free(sieve);
  }
  else
  {
    printf("Usage: perp <BASE> <MAXDIGITS>\n");
  }
}
```

- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

02-20-2021, 11:07 PM (This post was last modified: 02-23-2021 02:47 AM by Gerson W. Barbosa.) **Post: #24**

Gerson W. Barbosa
Senior Member

Posts: 1,369
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Valentin Albillo Wrote: (02-20-2021 12:32 AM)

Concoction the Sixth: Weird Year

Both *RPL* code and the resulting list of the years have been produced (without comments or explanation), but nearly all the questions have been left unanswered, i.e.:

- What is the *"simple but highly remarkable (striking, in fact) numeric property"* ?
- (a) How many years will be listed in the output ? ,

- (b) What will be the next predicted potentially catastrophic year after 2020 ?,
- (c) Should we be concerned ?

These were being answered, either explicitly or implicitly, in a post-scriptum to post #17 while you were still writing your post. Sorry for the delay.

> **Valentin Albillo Wrote:** (02-20-2021 12:32 AM)
>
> Finally, programs written in other than *RPL* would be welcome for variety and to let readers better understand what the code does and how their *RPN* calcs (say) would deal with the problem.

As you please, on the HP-75C:

```
10 OPTION BASE 1
15 DIM P(20),Y(100)
20 INPUT L
25 N=CEIL(3*(4*L/LOG(L)^2)^(1/3))
30 FOR I=1 TO 20
35 READ P
40 P(I)=P*P
45 NEXT I
50 C=0
55 FOR I=1 TO N
60 FOR J=1 TO N+1-I
65 S=0
70 FOR K=J TO I+J-1
75 S=S+P(K)
80 NEXT K
85 IF S<=L THEN C=C+1 @ Y(C)=S
90 NEXT J
95 NEXT I
100 FOR I=1 TO C-1
105 FOR J=I TO C
110 IF Y(I)>Y(J) THEN T=Y(I) @ Y(I)=Y(J) @ Y(J)=T
115 NEXT J
120 NEXT I
125 FOR I=1 TO C
130 DISP USING 135 ; Y(I);
135 IMAGE 5d
140 NEXT I
145 DISP
150 END
200 DATA 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47
,53,59,61,67,71

>RUN
?5000
    4    9   13   25   34   38   49   74
   83   87  121  169  170  195  204  208
  289  290  339  361  364  373  377  458
  529  579  628  650  653  662  666  819
  841  890  940  961  989 1014 1023 1027
 1179 1348 1369 1370 1469 1518 1543 1552
 1556 1681 1731 1802 1849 2020 2189 2209
 2310 2330 2331 2359 2384 2393 2397 2692
 2809 2981 3050 3150 3171 3271 3320 3345
 3354 3358 3481 3530 3700 3721 4011 4058
 4061 4350 4489 4519 4640 4689 4714 4723
 4727 4852 4899
>
```

Best regards,

Gerson.

Edited to fix lack of linguistic precision in the first paragraph

02-21-2021, 05:49 PM **Post: #25**

Nihotte(lma)
Member

Posts: 64
Joined: Mar 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (02-20-2021 12:32 AM)
>
> **Hi, all**:
>
> ...
>
> **Concoction the First: Weird limit**
>
> Point **d** is still unaddressed: ***d.*** *Can you explain the constant component of said [asymptotic] expression ?*
> ...
>
> As stated, **Last Chance**. Thanks and best regards.
> **V.**

I completed my initial result by calculating the rates of increase between the different successive results obtained.
If I consider each f(i) and if I calculate the Rate for i by (f(i) - f(i-1)) / (i - i-1), it will be :
f(1) = 2.71959
f(2) = 4.67827 ---> 1.95868 / (2 - 1) = 1.95868
f(2.021) = 4.71806 ---> 0.03979 / (2.021 - 2) = 1.89476
f(3) = 6.66808 ---> 1.95002 / (3 - 2.021) = 1.99185
f(pi) = 6.95027 ---> 0.28219 / (pi - 3) = 1.99297
f(4) = 8.66601 ---> 1.71574 / (4 - pi) = 1.99843
f(5) = 10.66641 ---> 2.0004 / (5 - 4) = 2.0004
f(5+1/6) = 10.99947 ---> 0.33306 / (5+1/6 - 5) = 1.99836
f(10) = 20.65914 ---> 9.65967 / (10 - (5+1/6)) = 1.99855
f(15) = 30.667 ---> 10.00786 / (15 - 10) = 2.001572
f(20) = 40.66927 ---> 10.00227 / (20 - 15) = 2.000454
and I can see that each time the ratio gives a result really near 2
Each time ? No !
There is one exception with (f(2.021) - f(2)) / (2.021 - 2) = 1.89476
Even, if you evaluate (f(3) - f(2))/ (3 - 2) = 1.98981, the result keeps a sign of this particularity
I can't say if it would possible to notice this lower result in other places that are independent or distant from 2.021
(excluding the portion before i=3, of course), but I can say that is already weird !

EMAIL PM FIND QUOTE REPORT

02-23-2021, 06:37 PM (This post was last modified: 02-23-2021 06:43 PM by Gerson W. Barbosa.) **Post: #26**

Gerson W. Barbosa
Senior Member

Posts: 1,369
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

#6

Longer, but faster: 17.547 seconds on the HP 50g (previously 29.952 seconds for argument = 5000)

```
« DUPDUP 4 * SWAP LN SQ / 3 XROOT 3 * CEIL { } 1 DUP2 ROT 5 ROLL
START NEXTPRIME + LASTARG NIP
NEXT DROP SQ DUP SIZE 2 SWAP
FOR i 1 OVER SIZE 1 + i -
FOR j DUP j DUP i + 1 - SUB ∑LIST 4 PICK OVER ≥ { ROT + SWAP } { DROP } IFTE
NEXT
NEXT + SORT REVLIST
WHILE DUP HEAD PICK3 >
REPEAT TAIL
END REVLIST NIP
»

Checksum: # 5E22h
Bytes: 238.5 bytes
```

This is closer to the HP-75C version, except for the prime list:

```
« DUPDUP 4 * SWAP LN SQ / 3 XROOT 3 * CEIL { } 1 DUP2 ROT 5 ROLL
```

```
START NEXTPRIME + LASTARG NIP
NEXT DROP SQ DUP SIZE 1 SWAP
FOR i 1 OVER SIZE 1 + i -
FOR j DUP j DUP i + 1 - SUB DUP SIZE 1 - NOT { 0 + } IFT ∑LIST 4 PICK OVER ≥ { ROT + SWAP } { DROP }
IFTE
NEXT
NEXT ROT DROP2 SORT
»

Checksum: # A346h
Bytes: 226.5 bytes
21.510 seconds
```

As a comparison, the BASIC program takes about 96 seconds on the HP-75C. Also, the BASIC program is limited to arguments up to 5492, unless of course the prime list is extended.

EMAIL PM FIND QUOTE REPORT

02-23-2021, 11:28 PM **Post: #27**

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Hi all,

Thank you very much for your overwhelming participation in my ***S&SMC#25***, much, much appreciated. I hope you've enjoyed its two main tasks: *sleuthing* your way through the challenges, using your brains and your math intuition, then *writing* efficient code for your favorite HP calcs to let them take away the drudgery. Seeing the many posted solutions and insights, it's quite clear you did well !

Now for my *original solutions*. Giving the solutions to *all* six *Concoctions* at once would make for a tremendously long post which would be most unwieldy to write and to read, so (as I did in some previous *S&SMC*) I'll give the solutions *one Concoction at a time*, beginning with the first one, immediately below: first my code and ***Sleuthing*** process, then my ***Results***, and finally my extensive ***Comments***. Last but not least, ***The Hall of Fame*** !

Note: My **HP-71B** code *may* use keywords from the *JPC ROM*, *MATH ROM*, *HP-IL ROM* and *STRINGLX LEX* file, executed on **go71b**, while *RPN* code is for the **HP-42S**, executed on a **DM42**.

My original solution for *"Concoction the First: Weird Limit"*

The Sleuthing

This simple *2-line, 96-byte* **HP-71B** program asks for the *sum* to exceed, then simulates *10, 100, ..., 100,000* tests and outputs the *average counts* :

```
1  DESTROY ALL @ INPUT N @ FOR L=1 TO 5 @ RANDOMIZE 1 @ T=10^L @ M=0 @ FOR I=1 TO T
2  S=0 @ WHILE S<N @ S=S+RND @ M=M+1 @ END WHILE @ NEXT I @ DISP T;M/T @ NEXT L

>RUN
? 1
      10      2.7            { error:  0.01828, see below }
      100     2.79           { error: -0.07172 }
      1000    2.729          { error: -0.01072 }
      10000   2.7084         { error:  0.00988 }
      100000  2.71959        { error: -0.00131 }
```

This simpler *24-step, 46-byte RPN* program accepts the sum and the number of tests and produces the average count to exceed it:

```
LBL "SUMRND"   SEED      X<Y?
"Wait..."      LBL 01    GTO 00
AVIEW          CLX       DSE ST Z
STO 01         LBL 00    GTO 01
X<>Y           ISG 00    RCL 00
 0             LBL 00    RCL/ 01
STO 00         RAN       CLD
E^X             +        END
```

```
1 [ENTER] 1E6 [XEQ] "SUMRND" -> 2.717405      { error: 0.0008768 }
1 [ENTER] 1E7 [XEQ] "SUMRND" -> 2.7180730     { error: 0.0002088 }
```

Running one or the other (whichever is faster), we get the following results:

N	S = **1**	S = **2**	S = **2.021**	S = **3**	S = **Pi**	S = **4**
10	2.7	4.8	4.8	6.6	7.1	9.1
100	2.79	4.72	4.75	6.81	7.13	8.85
1E3	2.729	4.701	4.737	6.701	6.978	8.708
1E4	2.7084	4.6751	4.7132	6.6757	6.9571	8.6718
1E5	2.71959	4.67827	4.71806	6.66808	6.95027	8.66601
1E6	2.717405	4.671500	4.712124	**6.666287**	6.949634	**8.664481**
1E7	**2.7180730**	**4.6707911**	**4.7118212**	-	**6.9500464**	-

N	S = **5**	S = **5+1/6**	S = **10**	S = **15**	S = **20**
10	11.2	11.5	20.9	30.6	41.1
100	10.77	11.22	20.66	30.72	40.82
1E3	10.719	11.07	20.623	30.604	40.68
1E4	10.673	11.0006	20.6876	30.7079	40.7027
1E5	**10.66641**	**10.99947**	20.65914	**30.66700**	**40.66927**
1E6	-	-	**20.664967**	-	-

The Results

Considering the data obtained above, these are my answers:

- ***a.*** *What do you think is the average count of generated random numbers for their sum to exceed* ***1*** *? Can you recognize what the exact value would be ?*

  The limit seems to be **~ 2.7181**, which I recognize as the constant ***e = 2.7182+***

- ***b.*** *What would the average count be for the sum to exceed* ***2*** *? To exceed* ***2.021*** *? To exceed* ***Pi*** *?*

  Count(**2**) = **~ 4.67079**, Count(**2.021**) = **~ 4.71182**, Count(**Pi**) = **~ 6.95004**

- ***c.*** *What do you think is the asymptotic expression for the average count needed to exceed large values of the sum ?*

  We have Count(**5**) = **~ 10.666**, Count(**10**) = **~ 20.665**, Count(**15**) = **~ 30.667**, Count(**20**) = **~ 40.669**

  so the *asymptotic expression* for large sums *S* seems to be: ***Count(S) ~ 2*(S + 1/3)***

- ***d.*** *Can you explain the constant component of said expression ?*

  The constant in the asymptotic expression seems to be **1/3**, and there's a formal explanation but I'll give here an easier, informal one: when the last random number added up results in a sum exceeding the given threshold, the overshoot is also uniformly distributed in *[0,1)*, and the expected value of this overshoot is the expected value of the *minimum* of *two* uniform random variables (the random number and the overshoot), which theoretically is **1/3** (i.e.: average of `MIN(RND,RND) = 1/3`).

  If you don't know that piece of theory,this modification of the above **HP-71B** program will produce it for a given sum by running *10-10,000* tests. It just simulates the process looking at the average overshoot (instead of the average of the count of random numbers generated), and computing the average of the fractional parts (overshoot) of the sum (for the case of integer thresholds), like this:

```
1  DESTROY ALL @ INPUT N @ FOR L=1 TO 4 @ RANDOMIZE 1 @ T=10^L @ M=0 @ FOR I=1 TO T
2  S=0 @ WHILE S<N @ S=S+RND @ END WHILE @ M=M+FP(S) @ NEXT I @ DISP T;M/T @ NEXT L

>RUN
?
```

?	S = **5**	S = **10**	S = **15**
10	0.356...	0.262...	0.289...
100	0.305...	0.321...	0.276...
1000	0.323...	0.335...	0.333...
10000	**0.333...**	**0.332...**	**0.333...**

The Comments

The limit average count for the sum of a series of *[0,1)* uniformly distributed random numbers to exceed **1** is exactly ***e* = *2.71828182845904523536+***, the base of the natural logarithms, which is pretty *"weird"* and can be considered an analog of *Buffon's Needle* experiment to estimate the value of ***Pi***. Here we don't throw needles on a grid but merrily add up random numbers keeping count and we get ***e*** instead.

These are the exact theoretical values for the sum to exceed *S*:

S	Average Count	20-decimal value
1	e	2.71828182845904523536
2	$e^2 - e$	4.67077427047160499187
3	$(2 e^3 - 4 e^2 + e)/2$	6.66656563955588990415
4	$(6 e^4 - 18 e^3 + 12 e^2 - e)/6$	8.66660449003269543723
5	$(24 e^5 - 96 e^4 + 108 e^3 - 32 e^2 + e)/24$	10.66666206862241185802

This is the general formula to numerically compute the theoretically exact value ...

$$f(x) = \sum_{k=0}^{[x]} (-1)^k \frac{(x-k)^k}{k!} e^{x-k}$$

where $[x]$ is the integer part of x.

... and this is my simple *1-line, 53-byte* **HP-71B** program to instantly compute them given the sum to exceed:

```
      1  DESTROY ALL @ INPUT X @ S=0 @ FOR K=0 TO IP(X) @ S=S+(K-X)^K/FACT(K)*EXP(X-
K) @ NEXT K @ DISP S

      >RUN
      ?        1      2.71828182846         2.021      4.71182750642
               2      4.67077427047            Pi      6.94950388760
               3      6.66656563953         5+1/6      11.0000029914
               4      8.66660448999
               5      10.6666620697
              10      20.6666664745
```

As the successive terms have *alternating signs* and go on increasing, the precision for large sums (say > **10**) *degrades* very quickly and we can see that for *Sum* > **100** the result is but *garbage*:

```
              100     -2.69702821806E43     { correct value: 200.666666666... }
```

Implementing the above exact formula in *RPN* (24 steps, 37 bytes) and running it in the 34-digit **DM42** we get the following:

```
      LBL "FSUMRN"   E^X          x
      STO 00         LASTX        +
      IP             +/-         DSE 01
      STO 01         RCL 01      GTO 00
       0             Y^X         RCL 00
      LBL 00         LASTX       E^X
      RCL 00          N!          +
      RCL- 01         /          END
```

A sample run would be:

```
      1 [XEQ] "FSUMRN"   ->    2.71828182846
```

and assorted results truncated to *20 decimal digits* (use `[SHOW]` to see them in full) would be:

```
     Sum   Average count               Sum     Average count
    -------------------------------------------------------------------
      1     2.71828182845 904523536    2.021    4.71182750642 763255399
      2     4.67077427047 160499187        e    6.10400234136 375415166
      3     6.66656563955 588990414       Pi    6.94950388752 954473480
      4     8.66660449003 269543722    5+1/6   11.00000299090 420020529
      5    10.66666206862 241185801   20+1/6   40.99999999999 999999992
     10    20.66666666647 631880061    20.21   41.08666666666 666666663
     15    30.66666666666 666034379   21+1/6   43.00000000000 000000000
     20    40.66666666666 666666648
```

which are fully correct to the digits shown. Though the precision attained using the 34-digit **DM42** is much greater than using the 12-digit **HP-71B** program, for large sums it will still quickly *degrade*. For instance:

```
   50   100.666667031        { only ~ 6 correct decimals, about 9 digits in all }
  100   1.12982914443E21     { utter garbage, should be 200.6666666666666666...}
```

The Hall of Fame

Some of you did bravely tackle this *Concoction the First: Weird Limit*, namely these four experts:

- **robve** posted code for the **HP Prime**, generic *BASIC* and *Python*, as well as correct results for questions ***a*** and ***b***, gave ***c*** a try and left ***d*** unanswered. A fine achievement which would be finer had he followed the rules re ony HP models and HP languages, as everyone else did.

- **ramon_ea1gth** posted *RPL* code and correctly guessed ***e*** as the limit.

- **Werner** posted *RPN* code with results identical to **robve**'s post, but correctly guessed the asymptotic expression.

- **Nihotte(lma)** posted *RPN* code for the **HP-35s** and *RPL* code for the **HP48G**, correctly guessed the limit to be ***e*** and stated that his results match those of **robve**. He also used a novel (and bold!) *thinking-out-of-the-box* attempt to guess the asymptotic expression using the **HP10BII+** to fit the data to a *linear regression*, getting correctly the *2*S* part, and later trying out yet another approach.

That's all for now, thanks a lot to those who contributed, I really hope you enjoyed it. I'll post my original solutions for ***"Concoction the Second: Weird Sum"*** in a couple of days. **Stay tuned !** :)

Best regards.
V.

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

PM WWW FIND EDIT QUOTE REPORT

02-24-2021, 10:27 AM **Post: #28**

EdS2
Member

Posts: 274
Joined: Apr 2014

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Very nice, thanks Valentin! I didn't play, but I enjoy watching the sport.

EMAIL PM FIND QUOTE REPORT

02-24-2021, 04:42 PM **Post: #29**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Valentin, nice result and in-depth investigation.

Sorry for this long reply:

As a kind suggestion and to offer some respectful constructive feedback: criticizing how we post our results is not helpful. Most of us don't have a lot of time to work on fun stuff. We cannot delay our posts to the end of the week, which is bad for two reasons: 1) it looks like we are just summarizing what other people already posted, and 2) it is not competitive to post our replies very late (because you hinted at some competition for working on these concoctions). If you want more participants and if you want everyone to post in a more organized way then simply do not hint at a competition and produce a Hall Of Fame outcome. It was fun to work on this, but I am not so sure I want to do this again.

Obviously, searching online or looking at other posts totally spoils the fun working on this, at least for me. All results and updates I posted are solely mine! Please note that I mentioned **e** in my post as the likely result for large trials. I gave no further explanation, as I started thinking about a theoretical result on the expected number of trials for a sum of random variables to exceed a threshold, but that knowledge is no longer at the top of my head, so I let it go and moved on to the other concoctions. Now that the first concoction has ended I verified my gut feeling about this. The result is known to converge to **e** as the expected value:

$$\mathbb{E}[X] = 1 + \sum_{k\geq 1} \Pr[X > k] = 1 + \sum_{k\geq 1} \frac{1}{k!} = e$$

where *X* is the number of trials you need for the sum to exceed 1. Indeed, a Buffons Needle-like or Monte Carlo approach to estimate **e**.

Please note that I posted my initial results early and added some new results as I went back to work on the concoctions. I think that most of us approach it that way, because "aha!" moments and inspiration are not constrained to a single day or hour or even a week, arriving with a sort of Gaussian distribution in our heads between your initial post and the deadline, rather than early or late. We also have a day job to take care of first and foremost.

I also thought it would be fine to have my posts combined into one post, deciding to update my initial post with **EDIT**, which seems reasonable and fair as it indicates what I added or changed. For example, once I went back to work on this I found I misunderstood one formula, corrected my program, and produced the result that you probably looked for so that felt satisfying. In that case it is easy to see that 1/A is the result, which works for any A not only 2021 because 2021 dominates the first primes in the series. Basically, the sum is approximately over terms x^n/(A^n+x^n) and converges to 1/A quickly when 1<x<<A. I don't need to program that further to understand it.

Also, why would you criticize posting extra code like BASIC and Python bad, **when posted in addition** to HP PRIME code? HP PRIME is not banned, which I had asked. If banned then that should be made more clear and I will no longer participate because I do not own a physical HP-71B or other vintage HP calculator (though I will be on the lookout for a used HP-71B that works when available at a reasonable price.)

To try and run my HP PRIME programs and learn more about the HP PRIME as I go, I spent hours typing them into the HP PRIME on that tiny key pad. I felt that using an virtual calculator on my laptop isn't what you meant these concoctions to be for, since **the intent is to actually use the calculators instead of letting them collect dust.**

On the subject of calculators collecting dust, the first calculator I used was a HP-45 in the 80s that my Dad owned and cherished. Not sure if he still actively uses it, but he still cannot part with his HP-45!

\- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

02-24-2021, 08:01 PM **Post: #30**

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

.
Hi, **robve**:

> **robve Wrote:** (02-24-2021 04:42 PM)
> Valentin, **nice result** and in-depth investigation.

Thanks for your appreciation and most especially for your participation and *very* comprehensive results.

> **Quote:**
> Sorry for this **long reply**:

Never mind, as far as I'm concerned *the longer, the better*. And this reply of mine isn't small fry, either.

> **Quote:**
> As a kind suggestion and to offer some respectful constructive feedback: criticizing how we post our results is not helpful.

Could you please give the exact post number of the particular post where I did such criticism ?

> **Quote:**
> Most of us don't have a lot of time to work on fun stuff. We cannot delay our posts to the end of the week, which is bad for two reasons: 1) it looks like we are just summarizing what other people already posted, and 2) it is not competitive to post our replies very late (because **you hinted at some competition** for working on these concoctions).

I understand the *lack-of-time* factor, but as for the *"you hinted at some competition for working on these concoctions"*, that's **not** so. This is *S&SMC #***25** and if you have a look at the previous 24 (I know, I know, you don't have the time) you'll see that they've *never* been posited as *"competitions"*. This one wasn't either.

> **Quote:**
> If you want more participants and if you want everyone to post in a more organized way then simply **do not hint at a competition** and produce a **Hall Of Fame** outcome.

As I've just explained, I **don't** hint at a competition and never have, that's your own idea, not mine.

And as for the *"Hall of Fame"*, it's a *novel* idea I had a few days ago (this is the very *first* time I've included it) and it was intended as a compliment and for showing my appreciation to those people who (like yourself) took the time and effort to post some results, **not** as a score-card or podium or something. Seeing your (over)reaction, perhaps it wasn't such a good idea after all so after posting my 5 remaining solutions (all of which do include their respective *"Hall of Fame"*, I'll drop it for good. Or not.

> **Quote:**
> It was fun to work on this, but **I am not so sure I want to do this again**.

It's up to you *(", New York, New York.")*. Anyway, thanks for participating in this one, much appreciated.

> **Quote:**
> Obviously, searching online or looking at other posts totally spoils the fun working on this, at least for me. **All results and updates I posted are solely mine!**

That final statement (with *exclamation point* and all) just reinforces my feeling that you're taking this humble challenge *much too* ***seriously***. It is and always has been intended for longer than a decade as ***fun***, *diversion*, never as a *competition*, that idea is of your own making, not mine. You know, this is not the IMO or Putnam Competition and I won't give you a Fields Medal either so please **relax**, take it easy !

> **Quote:**
> Indeed, a **Buffons Needle**-like or Monte Carlo approach to estimate **e**.

Yes. And a very simple one, even simpler than *Buffon*'s for Pi. It would be nice to come up with simple stochastic procedures to estimate other ubiquitous math constants.

> **Quote:**
> Please note that I posted my initial results early and added some new results as I went back to work on the concoctions. I think that most of us approach it that way, because "aha!" moments and inspiration are not constrained to a single day or hour or even a week, [...]. **We also have a day job to take care of first and foremost**.

I **insist**, *you're taking it too* ***seriously*** and *competitively*, and re that *"day job first and foremost"*, nobody is *forcing* you or *expecting* you to participate if you can't or won't. Think of this as solving today's paper *crosswords* or *Sudoku*: you do it when you have the time and if you feel like it. *Same here*.

> **Quote:**
> For example, once I went back to work on this I found I misunderstood one formula, corrected my program, and produced the result that you probably looked for so **that felt satisfying**.

See ? That's the idea, getting *satisfaction*. Not *stressing* over a nonexistent *"competitive"* edge.

> **Quote:**
> Also, why would you criticize posting extra code like BASIC and Python bad, **when posted in addition** to HP PRIME code?

Could you please give the exact post number of the particular post where I did such criticism ?

Quote:

On the subject of calculators collecting dust, the first calculator I used was a **HP-45** in the 80s that my Dad owned and cherished. Not sure if he still actively uses it, but he still cannot part with his HP-45!

The *HP-45* is also the second HP calculator I saw and handled (the first one was a much simpler, cheaper *HP-21*) and I admired it immensely, utterly state-of-the-art, classy, expensive-looking (and actually !), I so badly wanted one but couldn't afford it as the young adult I was back then. And its *hidden-timer* functionality was awesome in the extreme (if useless, for lack of a quartz crystal) ! ... Such fond remembrances ...

Thanks for taking the time (be careful, you now: that *day job* ...) to let me know what you think, Dr. Robert, really much appreciated, wish more people would do that instead of holding everlasting grudges.

In the next days I'll continue to post my original solutions to the remaining 5 *Concoctions* (which all include their respective *"Hall of Fame"*, sorry) and you may feel relieved to know that afterwards I won't post another *S&SMC* for a long, long time, if ever, so you'll have no problem in not participating.

Thanks again and best regards.
V.

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

PM WWW FIND EDIT QUOTE REPORT

02-25-2021, 12:37 AM **Post: #31**

PeterP
Member

Posts: 123
Joined: Jul 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

thank you Valentin for the wonderful explanations, can't wait to read the ones for the next "reveal".

For what its worth, robve, I have never felt these to be competitions, other than with myself. Most of us struggle with finding the time, though, I totally agree with that. These darn S&SMCs are just so much more fun than the stuff sitting in my inbox :-) (but - at least for me - they can take a lot of time...) I personally learned a lot from your post - I never knew that the Prime can do Python! - so thank you for posting and sharing. This is what makes this place so special, so much to learn from so many incredible people!

Cheers,

PeterP

PM FIND QUOTE REPORT

02-25-2021, 09:24 PM **Post: #32**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

PeterP Wrote: ⇒ (02-25-2021 12:37 AM)

For what its worth, robve, I have never felt these to be competitions, other than with myself. Most of us struggle with finding the time, though, I totally agree with that. These darn S&SMCs are just so much more fun than the stuff sitting in my inbox :-) (but - at least for me - they can take a lot of time...) I personally learned a lot from your post - I never knew that the Prime can do Python! - so thank you for posting and sharing. This is what makes this place so special, so much to learn from so many incredible people!

Yes, the challenges are great fun to work on and satisfying to crack. Valentin put a lot of hard work into them. Many kudos. I don't want to sound non-appreciative, which I am not. I certainly don't want to discourage anyone from participating! We are all helping each other to learn something, to improve our knowledge and skills. It is a win-win.

Just giving my feedback on wether or not one could expect comments on posted answers by the OP, which changes the competitive aspect of this IMO. That's all. No drama.

Cheers!

\- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

02-25-2021, 10:04 PM (This post was last modified: 02-25-2021 10:22 PM by PeterP.) **Post: #33**

PeterP
Member

Posts: 123
Joined: Jul 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Challenge 2, HP41, Sandbox module, unsophisticated, straight forward RPN.

```
Lbl 'AS
2021
STO 00 ;HP 41 is very slow with numbers
2
+
1/x
STO 01 ; First factor,
STO 02 ; Sum
2
St* 01 ;build second factor
RCL 00
3
STO 11 ;Current Prime
+
ST/ 01
GTO 05 ;calculate sum
LBL 06 ;find next prime
RCL 11 ; current prime
STO L
LBL 00 ;prime finding loop
LastX
2
+
PRIME?
GTO 01
GTO 00
LBL 01
View X ;show current prime
X <> 11 ; Store current prime
X <> 10 ; Put old current prime into last Prime
LBL 02
RCL 10 ; Last Prime
ST* 01
RCL 00
RCL 11
+
ST/ 01 ; update current factor
LBL 05 ; calc sum
RCL 01
ST+ 02
View 02 ; View Current Sum
GTO 06
```

The sum very quickly converges to 1/2021 or, to be more precise, after some sleuthing using different constants, to 1/A with A being the constant added in the denominator. Some sleuthing shows that its also irrelevant that the sum uses the prime numbers, one could use any sequence of numbers, even a constant 1 (ie sum of 1/(2021+1)^k).

doing a little bit more paper sleuthing on this, one can see that the numerator becomes a O(A^n) and the denominator O(A^(n+1)), meaning the sum trends to 1/A.

Cheers,

PeterP

PM FIND QUOTE REPORT

02-25-2021, 10:29 PM **Post: #34**

PeterP
Member

Posts: 123
Joined: Jul 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Challenge 5 - Perfect Primes.

I am clearly not understanding the challenge. Here is what I at first thought it meant:

> **Quote:**
>
> A perfect prime is a prime where there is no way you can make it composite by changing one digit (and one digit only) into any other digit.

So the brute force procedure to finding a perfect prime would be
1) Find a prime
2) Take the first digit of the prime
3) change it to all the numbers 0-9 in sequence
4) check each time if the resulting number is a prime number
5) proceed with 2) - 4) for all digits of the prime.
6) if none of the above creates a composite, you have a perfect prime.

Given that all numbers with a sum of the digits divisible by 3 are composite, it seems to me that the above test would always fail, as one could always change one digit up or down enough to make the sum of digits divisible by 3 and hence the number divisible by 3.

Would be great if someone could help me understand what I am missing (which I am sure is blatantly obvious)

Cheers,

PeterP

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

,
Hi, PeterP:

> **PeterP Wrote:** (02-25-2021 10:29 PM)
>
> Challenge 5 - Perfect Primes.
>
> I am clearly not understanding the challenge. Here is what I at first thought it meant:
>
> > **Quote:**
> >
> > A perfect prime is a prime where there is no way you can make it composite by changing one digit (and one digit only) into any other digit.
>
> [...]
> **Would be great if someone could help me understand what I am missing** (which I am sure is blatantly obvious)

With pleassure. You say you thought that it meant:

*"A perfect prime is a prime where there is no way you can make it **composite** by changing one digit (and one digit only) into any other digit."*

but actually it's exactly the opposite, it should be *"[...] there is no way you can make it **prime** [...]"*. Quoting my original definition:

> **I Wrote:**
>
> [...] consider a prime number so 'Perfectly Prime' (a PP for short, pronounced "Pepe") that ***changing any single digit would diminish its primeness by turning it into a composite number***. Note: We're talking about base-10 digits here.`

I hope that this makes it perfectly clear to you, and thanks for your interest and both recent messages you posted.

V.

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

02-26-2021, 12:52 AM **Post: #36**

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Hi all,

As I stated in my earlier post featuring my solution to ***"Concoction the First: Weird Limit"***, I'll give the solutions *one Concoction at a time*, and so here you are, my original solution for ***"Concoction the Second: Weird Sum"***, with the same *Sections* and structure, including ***The Hall of Fame*** !

Note: My **HP-71B** code *might* use keywords from the *JPC ROM*, *MATH ROM*, *HP-IL ROM* and *STRINGLX LEX* file, executed on **go71b**, while *RPN* code is for the **HP-42S**, executed on a **DM42**.

My original solution for *"Concoction the Second: Weird Sum"*

$$Sum = \sum_{k=1}^{\infty} \frac{2\,.\,3\,.\,5\,.\,7\,.\,11\,.\,13\,.\,\ldots\,.\; p_{k-1}}{(2021+\,2)(2021+\,3)(2021+\,5)\ldots(2021+\,p_k)}$$

What'so weird about this sum ?

The Sleuthing

I'll describe in detail a typical sleuthing procedure to try and answer the above question. First of all, the summation can be expanded as:

Sum *= 1/(2021+2) + 2/[(2021+2)*(2021+3)] + 2*3/[(2021+2)*(2021+3)*(2021+5)] + 2*3*5/[(2021+2)*(2021+3)*(2021+5)*(2021+7)] + ...*

= ***1/2023 + 2/4094552 + 6/8295562352 + 30/16823400449856 + ...*** *= 4.94804552195E-4 + ...*

and a quick glance convinces me that for ***2021*** the infinite summation converges very quickly and so I'll use a set of just 10 coefficients (the first 10 *primes*) to evaluate the sum, which will give more than 12 correct digits. This trivial *4-line, 104-byte* **HP-71B** program will do, let's run it:

```
1  DESTROY ALL @ INPUT N @ S=0 @ T=1 @ A=1
2  DATA 2,3,5,7,11,13,17,19,23,29
3  FOR I=1 TO 10 @ READ B @ T=T*A/(N+B) @ S=S+T @ A=B @ NEXT I
4  DISP S

>RUN
 ?   2021 [ENDLINE] -> 4.94804552201E-4
```

which is the correct 12-digit result. Now, *what's this value, can we* ***identify*** *it ?*. Well, let's edit *line 4* to use the **FRAC$** keyword to convert the *real* value to a *rational* approximation and run the program again: (**Note**: **FRAC$** is a keyword from the *JPC ROM*, if unavailable you can use my DEC2FRC subprogram to obtain a rational approximation):

```
4  DISP S;"= ";FRAC$(S)

>RUN
 ? 2021 [ENDLINE] -> 4.94804552201E-4 = 1/2021
```

so the sum's value is recognized as **1/2021**, which is quite unexpected and thus ***weird***. *Will this hold for values other than* ***2021*** *?*. Let's try some at random:

```
>RUN
?       2028 [ENDLINE] ->  4,93096646943E-4 =  1/2028
?       1007 [ENDLINE] ->  9.93048659383E-4 =  1/1007
?      -1357 [ENDLINE] -> -7.36919675757E-4 = -1/1357
```

and yes, *it indeed holds for other values* ! So, it seems the infinite sum using the set of primes will be equal to *the reciprocal* ***1/N*** of the given value ***N***, where *N* = ***2021*** in the original sum.

That using the set of *primes* we obtain the reciprocal ***1/N*** for other values of ***N*** is even ***weirder*** but, are we done ? What happens if we use sets *other* than the *prime* numbers ? Let's try this by *editing* the program to use **N = 2021** and accept *any* 10-element set from the user. The edited program looks like this *4-line, 111-byte* program:

```
1  DESTROY ALL @ OPTION BASE 1 @ DIM P(10)
2  S=0 @ T=1 @ A=1 @ N=2021 @ MAT INPUT P @ FOR I=1 TO 10
3  B=P(I) @ T=T*A/(N+B) @ S=S+T @ A=B @ NEXT I
4  DISP S;"= ";FRAC$(S)
```

Now let's `RUN` it with assorted sets, namely:

```
- The set of prime numbers   :  P(1)?  2,3,5,7,11,13,17,19,23,29  ->  4.94804552201E-4 = 1/2021
- The set of natural numbers:   P(1)?  1,2,3,4, 5, 6, 7, 8, 9,10   ->  4.94804552203E-4 = 1/2021
- The set of digits of Pi    :  P(1)?  3,1,4,1, 5, 9, 2, 6, 5, 3   ->  4.94804552202E-4 = 1/2021
- The set of all elements 0  :  P(1)?  0,0,0,0, 0, 0, 0, 0, 0, 0   ->  4.94804552202E-4 = 1/2021
- Any set of random numbers  :  P(1)?  RND,RND,RND, {7 more RND}    ->  4.94804552201E-4 = 1/2021
```

and for the *cherry on top*, this modified version (edits in **bold**) will let me demonstrate that the same result is produced when using arbitrary ***complex*** sets:

```
1  DESTROY ALL @ OPTION BASE 1 @ COMPLEX P(10),A,B,S,T
2  S=0 @ T=1 @ A=1 @ N=2021 @ MAT INPUT P @ FOR I=1 TO 10
3  B=P(I) @ T=T*A/(N+B) @ S=S+T @ A=B @ NEXT I
4  DISP S;"= ";FRAC$(REPT(S))

>RUN  ->  P(1)?  (1,3),(0,2),(-3,1),(2.1,1),(1,-2),(0.6,PI),(-3,1),(0,0),(1,5),(10,10)
      ->  (4.94804552201E-4, -1.488..E-18) = 1/2021
```

The Results

Considering all the data obtained above, I can summarize the results as follows:

- The value of the *original* infinite summation is ***4.94804552201E-4***, identified as **1/2021**.

What's so weird about this sum ?

- For the original summation (**N** = **2021**, set of ***primes***) the sum converges to **1/2021**, which is ***weird***.
- For ***any*** **N** (real or *complex* !) such that `ABS(N)>1`, the sum converges to **1/N**, which is ***weirder***.
- For *any* such **N**, the sum is ***independent of the set*** used (even if *complex*), which is ***weirdest***.

The Comments

The reason why this *infinite* summation produces the reciprocal **1/N** of a given **N** (not necessarily **2021**, as long as `ABS(N)>1`) can be better understood by considering the ***finite*** sum up to some index **K**, namely:

$$\sum_{k=1}^{K} \frac{a_1 a_2 \ldots a_{k-1}}{(x+a_1)\ldots(x+a_k)} = \frac{1}{x} - \frac{a_1 a_2 \ldots a_K}{x(x+a_1)\ldots(x+a_K)}$$

where the *right-hand side* is obtained by expanding the *left-hand side* sum into its separate terms and noticing that each term is of the form $A_{k-1} - A_k$ so that the sum becomes:

$$Sum = (A_0 - A_1) + (A_1 - A_2) + (A_2 - A_3) + \ldots + (A_{K-1} - A_K)$$

which is a ***telescoping*** *series*: all its terms *cancel out* except the ***first*** and the ***last***, i.e.: $Sum = A_0 - A_K$, where A_0 and A_K are as seen in the *right-hand side* of the above formula.

Now, this *finite* version of the sum does depend on the set of values a_1, a_2, ..., a_K used, as they do clearly appear in the *right-hand side* of the formula, but when we take it to the limit ***K -> Inf***, then the component $A_0 = 1/x$ in the *right-hand side* remains *intact* but the other component, which includes all the a_k coefficients, tends to ***0***, and thus the *infinite* sum becomes ***independent*** of the set of a_k used and we indeed have that

For all $a_1, a_2, \ldots$

$$\sum_{k=1}^{\infty} \frac{a_1 a_2 \ldots a_{k-1}}{(x+a_1)\ldots(x+a_k)} = \frac{1}{x}$$

The Hall of Fame

Again, some of you did bravely tackle this *Concoction the Second: Weird Sum*, namely these four experts:

- **robve** posted code for the **HP Prime** and got the correct sum, which he correctly identified, as well as the fact that using other **N** converges to **1/N**.

- **Werner** posted *RPN* code and also got the correct sum and the correct identification.

- **J-F Garnier** correctly identified the sum and also discovered that other values **N** converged to **1/N** and that using the *natural* numbers instead of the *primes* also gave the same sum.

- **PeterP** posted *RPN* code and correctly identified the sum, and also that using another **N** converged to **1/N** and that using the *prime* numbers is irrelevant and one could use *any* sequence of numbers, even a constant **1**.

That's all for now, as always thanks a lot to those who contributed, I really hope you enjoyed it. I'll post my original solutions for ***"Concoction the Third: Weird Integral"*** in a couple of days. **Stay tuned !**

- ***Note***: *For any comments not directly related to the math or code here but to ancillary matters such as this or that opinion on the rules or "Halls of Fame" or whatever, please* ***PM*** *me instead of posting them here. Let's keep this thread strictly mathematical and algorithmical in nature.* ***Thanks***.

Best regards.
V.

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

PM WWW FIND EDIT QUOTE REPORT

02-26-2021, 08:31 AM **Post: #37**

PeterP
Member

Posts: 123
Joined: Jul 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Thank you Valentin for your kind help, yet to my embarrassment I am still confused.

> **Valentin Wrote:**
> [...] consider a prime number so 'Perfectly Prime' (a PP for short, pronounced "Pepe") that ***changing any single digit would diminish its primeness by turning it into a composite number***. Note: We're talking about base-10 digits here.`

In another post (apologies for cheating) someone listed the first perfect prime according to their analysis to be 294001.

Now, let me ***change a single digit and diminish its primeness by turning it into a composite number***

29400**3** which is not prime as it is divisible by 3.

Now, it is entirely possible that the posted solution of 294001 as a perfect prime is incorrect. But it seems to me that I could change ***a single digit*** of any prime number in such a way that the resulting number is divisible by 3. Which would mean there are no perfect primes.

Maybe someone can help me understand how 294001 is a perfect prime (or give me any perfect prime and help me understand how and why it is a perfect prime and can't be made composite by changing any single digit.

Thank you so much for your indulgence in helping me understand!

Cheers,

PeterP

PM FIND QUOTE REPORT

02-26-2021, 08:39 AM **Post: #38**

Werner
Senior Member

Posts: 591
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **PeterP Wrote:** (02-26-2021 08:31 AM)
>
> > **Valentin Wrote:**
> > [...] consider a prime number so 'Perfectly Prime' (a PP for short, pronounced "Pepe") that ***changing any single digit would diminish its primeness by turning it into a composite number***. Note: We're talking about base-10 digits here.`
>
> Now, it is entirely possible that the posted solution of 294001 as a perfect prime is incorrect. But it seems to me that I could change ***a single digit*** of any prime number in such a way that the resulting number is divisible by 3. Which would mean there are no perfect primes.

Yes, you can change *a* single digit of any prime to make it divisible by 3.
In a perfect prime, however, you could change ANY digit, and it would not be prime anymore.

Take 11, for instance.
If you change its last digit to 2, it is no longer prime. But if you change it to 3, it IS prime, so 11 is not a perfect prime.
The goal is to find primes so that changing any digit would NOT result in another prime.

Cheers, Werner

EMAIL PM FIND QUOTE REPORT

02-26-2021, 11:37 AM **Post: #39**

PeterP
Member

Posts: 123
Joined: Jul 2015

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Werner Wrote:** (02-26-2021 08:39 AM)
>
> Take 11, for instance.
> If you change its last digit to 2, it is no longer prime. But if you change it to 3, it IS prime, so 11 is not a perfect prime.
> The goal is to find primes so that changing any digit would NOT result in another prime.
>
> Cheers, Werner

LIGHTBULB (i think). In other words, i cannot transform a perfect prime into another prime by changing one digit of it.
So if I change any digit of 294001 I will never get another Prime. (Hopefully I got it) Thank you Werner!

Cheers,

PeterP

PM FIND QUOTE REPORT

02-28-2021, 03:18 AM (This post was last modified: 03-01-2021 12:09 AM by Valentin Albillo.) **Post: #40**

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Hi all,

These are my *original solutions* for both ***"Concoction the Third: Weird Integral"*** and ***"Concoction the Fourth: Weird Graph"***, once more with the same *Sections* and structure, including ***The Hall of Fame***. For my previously posted solutions you can follow these links: ***"Concoction the First: Weird Limit"*** and ***"Concoction the Second: Weird Sum"***.

Note: My **HP-71B** code *might* use keywords from the *JPC ROM*, *MATH ROM*, *HP-IL ROM* and *STRINGLX LEX* file, executed on **go71b**, while *RPN* code is for the **HP-42S**, executed on a **DM42**.

My original solution for *"Concoction the Third: Weird Integral"*

$$\int_1^{\varphi} \frac{\Gamma\, ln(\varphi^2 - x)}{\Gamma\, ln\, x\, + \Gamma\, ln(\varphi^2 - x)}\, dx$$

What's so weird about this integral?

The Sleuthing

As before, I'll describe a typical sleuthing procedure to try and answer the above question. First of all, let's compute the integral's value to full 12-digit accuracy using the **HP-71B**, right from the command line:

```
>DESTROY ALL @ P=(1+SQR(5))/2 [ENDLINE]
>INTEGRAL(1,P,0,GAMMA(LN(P^2-IVAR))/(GAMMA(LN(IVAR))+GAMMA(LN(P^2-IVAR)))) [ENDLINE]

          .309016994376
```

which is the correct 12-digit result within *1 ulp*. In case you didn't recognize it, use my **IDENTIFY** subprogram (a must for everyone's *HP-71B* library !) which will identify it as ***1/(2*φ)*** (or an equivalent expression). Now let's create a *user-defined* function for the ***f(x)*** under the integral sign, to experiment a little. This trivial program will do:

```
1  DESTROY ALL @ P=(1+SQR(5))/2
2  DEF FNF(X)=GAMMA(LN(X))
3  DISP INTEGRAL(1,P,0,FNF(P^2-IVAR)/(FNF(IVAR)+FNF(P^2-IVAR)))

>RUN
      .309016994376
```

Now I'll find the value of ***f(x)*** and ***f(φ²-x)*** at the endpoints of the integration interval:

```
>SFLAG -1 @ FNF(1),FNF(P) @ FNF(P^2-1),FNFP(P^2-P) @ CFLAG -1

      9.99999999999E499      1.84069978531
      1.84069978531          9.99999999999E499
```

and we notice that ***f(1) ~ Inf***, which will do no harm, and that the values at the interval's endpoints for ***f(x)*** and ***f(φ² - x)*** are *symmetric*. Also, from now on we'll use the fact that $\varphi^2 = 1 + \varphi$.

Experimenting a little, we might compute the integral for other intervals, to see if there's some *invariance* or anything recognizable, but to no avail. Now, what if we try *other functions* using the *same* arguments ? This edited version of the above code *(94 bytes)* will help us test this idea with assorted *user-supplied* ***f(x)***:

```
1  DESTROY ALL @ @ P=(1+SQR(5))/2 @ INPUT "f(x)? ";F$
2  DEF FNF(X)=VAL(F$)
3  DISP INTEGRAL(1,P,0,FNF(P^2-IVAR)/(FNF(IVAR)+FNF(P^2-IVAR)))

>RUN
      f(x)?   GAMMA(LN(X))         ->   .309016994376
      f(x)?   GAMMA(X)             ->   .309016994375
      f(x)?   SQR(LN(X))           ->   .309016994375
      f(x)?   SQR(X)               ->   .309016994375
      f(x)?   SIN(X)               ->   .309016994375
      f(x)?   X^X/COSH(2.021*X)    ->   .309016994375
      f(x)?   SIN(SINH(X)+LN(X))   ->   .309016994375
      f(x)?   X^3-6*X-2            ->   .309016994375
      f(x)?   X                    ->   .309016994375
      f(x)?   1                    ->   .309016994375
```

and it's pretty clear what's happening, so no more sleuthing's needed, let's just go for the results.

The Results

After all the sleuthing above, I can summarize the results as follows:

- The numeric value of the integral is ***.309016994376*** (correct to 12 digits within *1 ulp*), identified as ***1/(2*φ)*** *{ or also (φ-1)/2 }*

- The value of this definite integral is ***independent of the function*** being integrated (as long as the same interval and arguments are used, ***f(x)*** is continuous and the integral's denominator isn't ***0*** inside the interval). This is **weird !**

The Comments

A key fact is to notice that φ^2 equals $1 + \varphi$, so the integral becomes:

$$I = \int_1^{\varphi} \frac{\Gamma\ ln(1+\varphi-x)}{\Gamma\ ln(x) + \Gamma\ ln(1+\varphi-x)}\ dx$$

and as we also saw above that the values at the endpoints for ***f(x)*** and $f(\varphi^2 - x)$ (i.e. $f(1 + \varphi - x)$) are symmetric, we perform the change of variable $z = 1 + \varphi - x$, $dz = -dx$, which after trivial algebraic manipulations turns the integral into the form:

$$I = \int_1^{\varphi} \frac{\Gamma\ ln(z)}{\Gamma\ ln(z) + \Gamma\ ln(1+\varphi-z)}\ dz$$

and as ***z*** is a dummy integration variable, we formally substitute it for ***x***, and the integral becomes now:

$$I = \int_1^{\varphi} \frac{\Gamma\ ln(x)}{\Gamma\ ln(x) + \Gamma\ ln(1+\varphi-x)}\ dx$$

which allows for adding this integral to the original one and then we have:

$$2I = \int_1^{\varphi} \frac{\Gamma\ ln(1+\varphi-x)}{\Gamma\ ln(x) + \Gamma\ ln(1+\varphi-x)}\ dx + \int_1^{\varphi} \frac{\Gamma\ ln(x)}{\Gamma\ ln(x) + \Gamma\ ln(1+\varphi-x)}\ dx$$

$$= \int_1^{\varphi} \frac{\Gamma\ ln(x) + \Gamma\ ln(1+\varphi-x)}{\Gamma\ ln(x) + \Gamma\ ln(1+\varphi-x)}\ dx = \int_1^{\varphi} 1\ dx = \varphi - 1 \quad \rightarrow \quad I = \frac{\varphi-1}{2}$$

and we get the same numerical value we computed earlier, now in symbolic form as $(\varphi-1)/2 = 1/(2*\varphi) =$ ***.309016994375***.

Note that this works for *any* ***f(x)*** (subject to the aforementioned limitations) because the expressions in the *numerator* and *denominator* in the sum above *cancel out* and you end up computing the integral of the *constant* function **1**, regardless of the ***f(x)*** originally used.

Other intervals $[a, b]$ and arguments $(x + u, w - x)$ are possible as long as $w - u = a + b$. Here we had $a = 1$, $b = \varphi$, $u = 0$ and $w = \varphi^2 = 1 + \varphi$.

The Hall of Fame

This time the experts which dared to deal with this *Concoction the Third: Weird Integral* are the following four people:

- **J-F Garnier** posted *RPN* code for the **HP-32S** and said that he successfully computed the numeric value to at least 11 places and had (presumably) deduced what's weird about the integral and its symbolic value as well, though he didn't post any results to avoid spoiling the fun for others.

- **robve** computed the integral using the **HP Prime** but alas, the result he posted is wrong. He also posted a *BASIC* program for the *SHARP PC-1350* which produced the *same* wrong result. Anyway, thanks for trying ...

- **Werner** posted *RPN* code for the **HP-42S** to be used with the built-in integration functionality but he didn't post any results, presumably to avoid spoiling the fun, as J-F did. In a subsequent message he explained his sleuthing and gave both the answer to the *"What's weird"* question as well as the correct symbolic value.

- **Nihotte(lma)** posted an *RPN* program for the **HP35s** and correctly computed the integral's numeric value, which he also identified symbolically as well.

My original solution for *"Concoction the Fourth: Weird Graph"*

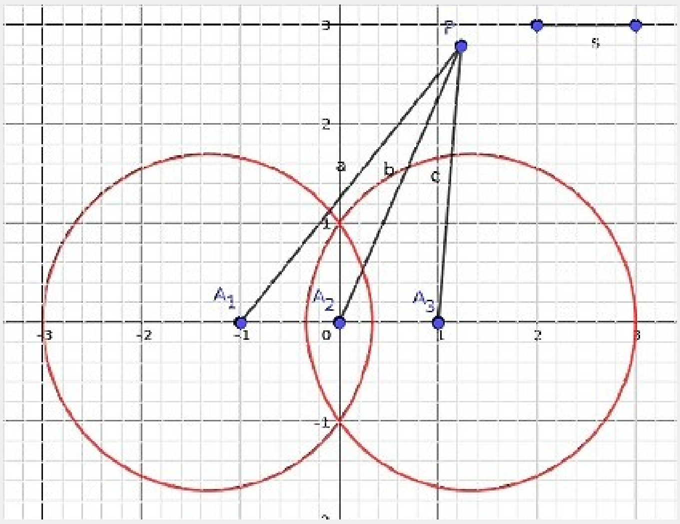
"Consider the following polynomial in two real variables ***x, y****:*

$$\boldsymbol{P(x, y)} = 9 x^8 + 9 y^8 + 36 x^2 y^6 + 54 x^4 y^4 + 36 x^6 y^2 - 100 x^6 - 4 y^6 - 108 x^2 y^4 - 204 x^4 y^2 + 182 x^4 - 10 y^4 - 84 x^2 y^2 - 100 x^2 - 4 y^2 + 9$$

Plot the graph of ***P(x, y) = 0. What's so weird about the graph ?****"*

The Sleuthing

In this case, the first thing to do is, well, to plot the graph of ***P(x, y) = 0***. As I stated in my *OP*, I won't post code or manual operations as I don't own any graphing calculators but I'll give the resultant *graphic* you should get, which comes out like this (**Note**: ignore the line segments and labels for now, they're explained in the ***Comments*** below):



And we can see that the graph is perfectly *symmetrical* respective of both *X,Y* axes and seems to consist of *two intersecting circles* (in ***red***, the *"crossed eyes"*) and *two isolated points* (*"the pupils"*), ***(-1, 0)*** and ***(1, 0)***, labeled as $\boldsymbol{A_1}$ and $\boldsymbol{A_3}$, respectively.

The graph in itself is funny-looking (thus *weird*) but is that all there's to it ? Let's explore the isolated points by looking at the intersections of ***P(x, y) = 0*** with the *X* axis, i.e. by looking at the roots of ***P(x, 0) = 0***, which we obtain by factorizing that polynomial:

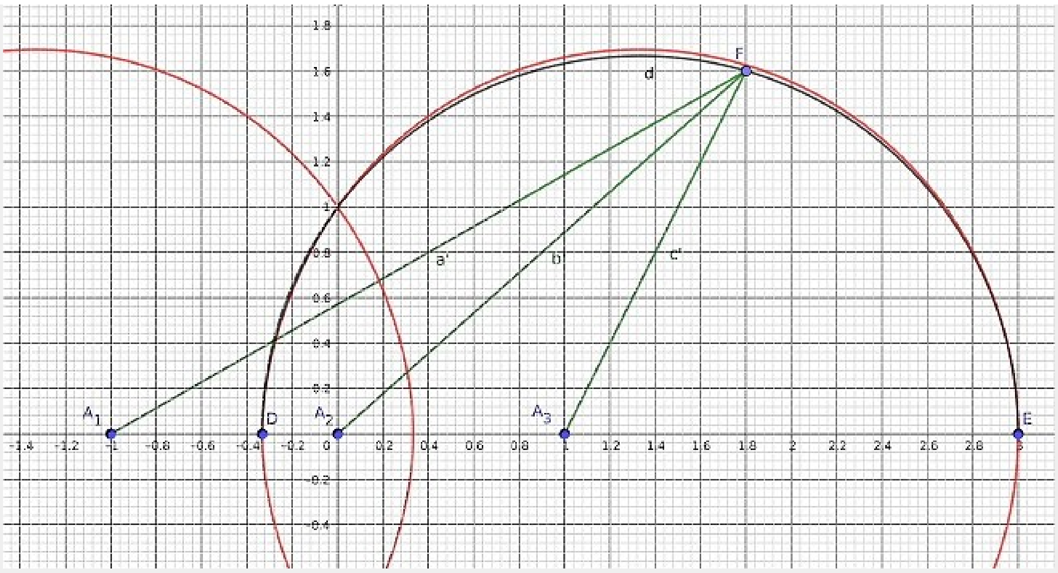
$\boldsymbol{P(x, 0)} = \boldsymbol{9 x^8 - 100 x^6 + 182 x^4 - 100 x^2 + 9} = 0$, which factorizes as $(x + 3)(3 x + 1)\boldsymbol{(x + 1)^2}(x - 3)(3 x - 1)\boldsymbol{(x - 1)^2} = 0$,

and we see that the graph intersects the *X* axis at the points *(-3,0), (-1/3, 0), (3, 0), (1/3, 0)* and also ***(-1, 0), (1, 0)*** which both turn out to be *double roots*, so each of them is a ***double isolated point*** of the graph. Their isolation means that there are no other graph points in their vicinity, which makes them *doubly weird*, thus ***weirder***.

Are we done ? *Not yet*. The *"crossed eyes"* part of the graph certainly seems to consist of two perfect circles at first sight but, *is that so ?* ***Are they true circles ?***. The question can be answered by *factorizing* ***P(x, y)*** and checking whether there are two factors which can be identified as the respective circles' equations, e.g.:

$Q(x, y) = 3 x^4 + 3 y^4 + 6 x^2 y^2 - 6 x^2 - 10y^2 + 3$ factorizes as $3 (x^2 + y^2 + 2\sqrt{3}/3\, y - 1) (x^2 + y^2 - 2\sqrt{3}/3\, y - 1)$,

and both 2^{nd}-degree polynomial factors are *equations of circles* so the graph of *Q(x, y) = 0* is the union of two true circles. Regrettably, our ***P(x, y)*** does ***not*** factorize that way so let's zoom in the first quadrant (the others are symmetrical) at suitably high magnification, and we get this view, where the graph of ***P(x, y)*** appears in ***red*** and a superimposed true circle appears in ***black***:



As you can see, they don't exactly match, *close but no cigar !* This means that the *"crossed eyes"* ara ***not*** true circles in fact, and that is most *unexpected* and thus ***weirdest***. Enough sleuthing, time for the results.

The Results

Once the sleuthing is over, I can summarize the results as follows:

- The graph consists of what seems to be *two intersecting circles* (the *"crossed eyes"*, which is *weird*), and *two* ***double isolated*** *points* (*"the pupils"*) at ***(-1, 0)*** and ***(1, 0)***, which is *weirder*.

- Despite initial impressions, the *"crossed eyes"* ***aren't*** true circles after all but only *very* close approximations, which is ***weirdest***.

The Comments

This polynomial ***P(x, y)*** is obtained as the *extended locus* of all points ***P*** such that the *signed sum* of the lengths of the three segments ***a***, ***b***, ***c*** from the three fixed points A_1 = ***(-1, 0)***, A_2 = ***(0, 0)*** and A_3 = ***(1, 0)*** to the point **P** is equal to the length of segment ***s***, here equal to ***1*** (see the first graph above). It is a generalization of the case for just a single point A_1, where the locus would be a circle of radius ***1***.

For the case of three arbitrary, distinct fixed points A_1, A_2, A_3, the resulting polynomial is always of degree ***8***, like here, except in degenerate cases. With our fixed points A_1, A_2 and A_3, the *"pupils"* are *double isolated points* but for other fixed points and/or lengths of ***s*** the double isolated points (the *"pupils"*) may actually *widen* to small *ovals* inside the bigger ovals (*"crossed eyes"*).

Also, we saw that the *"crossed eyes"* ovals weren't actually perfect circles by zooming in, but we can check it numerically with ease. For instance, let's assume that the right oval is a circle (see the second graph above). Points ***D*** = ***(-1/3, 0)*** and ***E*** = ***(3, 0)*** belong to the oval so if we assume it's really a circle then it would have center ***(4/3, 0)*** and radius ***5/3***. So far, so good.

But now let's consider point ***F = (9/5, 8/5)***, which lies in the true circle (superimposed in ***black***) because $\sqrt{(9/5 - 4/3)^2 + (8/5)^2} = 5/3$, but it doesn't lie in the *extended locus* (in ***red***) because the *signed sum* of the three segments from the fixed points to ***F*** is ***~ 0.97***, less than ***1.00*** (the length of segment ***s***), which is close but not exact, a ***~ 3 %*** error. Computing the differences for a suitable number of points on the true circle reveals that the maximum error is always below ***~ 3.5 %***, so though not exactly, the *"crossed eyes"* ovals are each *very* close to being a true circle.

The Hall of Fame

This time the experts which boldly addressed this *Concoction the Fourth: Weird Graph* are the following two people:

- **robve** used the **HP Prime** to plot the graph and gave the correct center for the *"circles"* but the wrong diameter. He also gave the correct coordinates for the *"pupils"* and remarked that they were quite difficult to compute to single points.

- **C.Ret** posted a very detailed account of his thorough sleuthing process, with graphs aplenty as well as algebraic disquisitions, getting right the *double isolated points* coordinates and characterization, and even tried to factorize the ***P(x,y)*** polynomial to no avail, stopping shy of recognizing that this meant the seemingly perfect circles actually weren't. A little zooming in would have settled the matter for good.

That's all for now, many thanks to those who contributed, I sure expect you enjoyed it. I'll post my *original solutions* for both ***"Concoction the Fifth: Weird Primes"*** and ***"Concoction the Sixth: Weird Year"*** in a few days. **Stay tuned !**

- ***Note***: *For any comments not directly related to the math or code here but to ancillary matters such as this or that opinion on the rules or "Halls of Fame" or whatever, please* ***PM*** *me instead of posting them here. Let's keep this thread strictly mathematical and algorithmical in nature. Thanks.*

Best regards.
V.

P.S.: Edited to include a "missing" expert in Concoction's Three's "Hall of Fame". Sorry for that !

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

PM WWW FIND EDIT QUOTE REPORT

02-28-2021, 10:35 AM (This post was last modified: 02-28-2021 10:37 AM by C.Ret.) **Post: #41**

C.Ret
Junior Member

Posts: 47
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Ah! Ah!

Many Thanks to Valentin for this good puzzle and the good explanation and pointing out the fact that the graph is not exactly a circle.

I didn't notice it and was wandering why I was unable to factorize Po(x,y) even with the correct equation of the expected, but outfitted circle equation.

By the fact, checking whenever the graph is or isn't a true circle is so easy by entering its equation in the Advanced Graphic Application of the HP Prime and zoom enough to get a good view of the disparity:

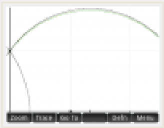


Fig.: Po(x,y) black Circle green.

How Can I have miss verifying this weird fact ?

PM FIND QUOTE REPORT

02-28-2021, 11:32 AM (This post was last modified: 02-28-2021 12:09 PM by J-F Garnier.) **Post: #42**

J-F Garnier
Senior Member

Posts: 484
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Now that the challlenge is closed, let me add some comments:

> **Valentin Albillo Wrote:** (02-28-2021 03:18 AM)
>
> **My original solution for *"Concoction the Third: Weird Integral"***
>
> $$\int_1^{\varphi} \frac{\Gamma \ln(\varphi^2 - x)}{\Gamma \ln x + \Gamma \ln(\varphi^2 - x)}\, dx$$
>
> ***What's so weird about this integral?***

> **Quote:**
>
> ***The Sleuthing***
>
> As before, I'll describe a typical sleuthing procedure to try and answer the above question. First of all, let's compute

the integral's value to full 12-digit accuracy using the **HP-71B**, right from the command line:

```
>DESTROY ALL @ P=(1+SQR(5))/2 [ENDLINE]
>INTEGRAL(1,P,0,GAMMA(LN(P^2-IVAR))/(GAMMA(LN(IVAR))+GAMMA(LN(P^2-IVAR)))) [ENDLINE]
```

First, would you have used the Enhanced Math ROM, you could have just typed:

```
>INTEG(1,P,0,GAMMA(LN(P^2-IX))/(GAMMA(LN(IX))+GAMMA(LN(P^2-IX)))) [ENDLINE]
```

After all, you are the one who suggested the use of such shortcuts

As I mentioned in my solution, the first aspect I noticed was the speed; I wasn't expecting to get a fast answer on a physical calculator of the 32S class for such complicate-looking equation combining the gamma and log functions.
I also noticed that the speed and the answer were independent of the target accuracy (set by the display mode on the 32S - from FIX 1 to FIX 11). That was really weird.
BTW, using a slow machine sometimes highlights some aspects that can be missed on faster systems such as Free42.

Moving to the HP-71/Emu71, I investigated more:

```
30 Y=0 @ A=0 @ F=(1+SQR(5))/2
60 DEF FNF(X)
70 A=GAMMA(LN(F*F-X)) @ Y=A/(GAMMA(LN(X))+A)
90 PRINT X;Y
100 FNF=Y
110 END DEF
130 E=.0001
140 DISP INTEGRAL(1,F,E,FNF(IVAR))
150 END

>RUN
1.30901699438 .500000000007
1.09656781074 .169413871978
1.52146617801 .830586128022
1.19554981675 .328227901097
1.422484172 .671772098903
1.02655614795 4.82413106951E-2
1.5914778408 .951758689306
```

It turned out that only 7 samples were used, whatever was the target accuracy set by the E value.
Having some knowledge of the Romberg algorithm, I recognized the first sample to be the midpoint, then the next two samples used for the 1st estimation and the last four samples used for the 2nd estimation.
It was quickly obvious that the sample pairs (2-3, 4-5, 6-7) gave values that sum to exactly 1.
That gave me the key for the explanation of the speed: the summation for any sample set was the same (and the exact one), so the algorithm reached its termination condition after the first two estimations.
Extrapolating to a single sample at the midpoint, I got the symbolic answer: (φ-1) x 1/2

This is of course a consequence of the symmetry of the function to integrate as Werner and Valentin explained.
It is interesting to note that my observations had also to do with the HP implementation of the Romberg algorithm: although the samples are non-uniform, they are symmetric relative to the midpoint, and that gave the effect I reported.

Thanks for the challenge that was at the intersect of my interests for math puzzles and numerical algorithms (such as used in HP machines).

J-F

EMAIL PM WWW FIND QUOTE REPORT

02-28-2021, 02:21 PM **Post: #43**

Albert Chan
Senior Member

Posts: 1,290
Joined: Jul 2018

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Werner Wrote:** (02-18-2021 10:28 AM)
>
> The key to the value of the integral in #3 is to realize that the integral is symmetric in the integral boundaries.
> Basically it is
>
> integral(a,b,f(a+b-x)/(f(x) + f(a+b-x)),dx)

Without noticing the symmetry, if we "fold" the integral, we get the same result.

$$\int_1^{\varphi} \frac{\Gamma\, ln(\varphi^2 - x)}{\Gamma\, ln\, x + \Gamma\, ln(\varphi^2 - x)}\, dx$$

Let f(x) be integrand of above integral.

I = ∫(f(x), x = 1 .. φ)
  = ∫(f(x) + f(1+φ-x), x = 1 .. φ) / 2
  = ∫(f(x) + f(φ²-x), x = 1 .. φ) / 2
  = ∫(1, x = 1 .. φ) / 2
  = (φ-1)/2

03-01-2021, 02:19 AM (This post was last modified: 03-02-2021 05:03 PM by robve.) **Post: #44**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Some additional comments on concoction five.

First, the HP PRIME HPPL program with the Sieve of Eratostenes is the smart way to approach this problem efficiently. Unfortunately, sieving for primes above 20,000 or so becomes tricky, due to HPPL list length restrictions to 20,000. One could populate the list with 64 bit integers and use bit banging to increase the prime sieving space 64 fold, which is just enough to find the first 5 perfect primes.

Without sieving is straightforward to implement with a few loops, but slow and takes a few minutes rather than seconds (e.g. compared to C). The program hunts for perfect primes within designated bounds B and E:

```
EXPORT SLOWPP()
BEGIN
  LOCAL B,D,E,F,I;
  LOCAL J,K,P,Q;
  // BEGIN SEARCH AT B
  B := 11;
  // END SEARCH AT E
  E := 9999999;
  FOR P FROM B TO E STEP 2 DO
    IF isprime(P) THEN
      M := FLOOR(LOG(P));
      F := 1;
      FOR J FROM 0 TO M DO
        I := 10^J;
        // Q = P WITH JTH DIGIT SET TO 0
        Q := FLOOR(P/I/10)*I*10+ROUND(FP(P/I)*I);
        // D = JTH DIGIT OF P (0 to 9)
        D := FLOOR(P/I) MOD 10;
        FOR K FROM 0 TO 9 DO
          IF K <> D AND isprime(Q+K*I) THEN
            F := 0;
            BREAK;
          END;
        END;
        IF F=0 THEN
          BREAK;
        END;
      END;
      IF F THEN
        PRINT("PERFECT "+P);
      END;
    END;
  END;
END;
```

Hence, hereby my alternative submission in HPPL.

Second, I stated that theoretically more perfect primes should exist for bases smaller than 10 and also gave a list of the first base-perfect primes for base 2 to 20. Running additional experiments corroborates this claim empirically verified for primes up to 1,000,000,000 (with a small modification made to the C program):

```
base | PP count |
2    | 7179981  |
3    | 10070244 |
4    | 1521240  |
5    | 7627392  |
6    | 12456    |
7    | 3762873  |
8    | 98926    |
9    | 400411   |
10   | 3167     |
11   | 1018816  |
12   | 37       |
13   | 558553   |
14   | 403      |
15   | 2821     |
16   | 743      |
17   | 153894   |
18   | 0        |
```

An interesting pattern emerges, which shows something else, besides the frequency of base-perfect primes is higher for smaller bases as expected. Note that prime base perfect primes are more prevalent.

EDIT
Try it yourself: my C program to find PPs for any base for the given max:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char **argv)
{
  if (argc > 2)
  {
    long i, j, k;
    long base = atoi(argv[1]); // perfect prime base
    double lnbase = log(base);
    long max = atoi(argv[2]);  // primes up to max (exclusive)
    long maxd = 0;             // primes of maxd digits
    long num = 0;              // number of perfect primes found
    long len = 0;              // length of the array
    char *sieve;               // the sieve array (0 = composite, 1 = prime)

    // make max odd, e.g. 1000000000 -> 99999999 has 9 digits to perturb, not 10
    max -= !(max & 1L);

    // len = base^maxd for maxd>0 such that len >= max
    do
      len = pow(base, ++maxd);
    while (len < max);

    sieve = (char*)malloc(len);

    // init sieve, we should keep only odd values
    for (i = 0; i < len; ++i)
      sieve[i] = (i & 1L);

    // sieve for primes
    for (i = 3; i < len; i += 2)
    {
      while (i < len && sieve[i] == 0)
        ++i;
      for (j = 2*i; j < len; j += i)
        sieve[j] = 0;
    }

    // sieve for perfect primes
    for (i = 3; i < len; i += 2)
    {
      while (i < len && sieve[i] == 0)
        ++i;
```

```
      if (i > max)
        break;
      if (i < len)
      {
        long m = floor(log(i)/lnbase); // m = number of digits of i
        long p = 1;                    // p = base^j
        char h = 1;                    // h = 1 if i is perfect
        // for each jth digit in prime i, from least to most significant
        for (j = 0; j <= m; ++j, p *= base)
        {
          // q = prime i with jth digit zero
          long q = i%p + i/p/base*p*base;
          // d = jth digit of prime i
          long d = i/p%base;
          // twiddle jth digit and check for prime
          for (k = 0; k < base; ++k)
            if (k != d && sieve[q+k*p])
              h = 0;
        }
        if (h)
          ++num;
      }
    }

    printf("num=%ld\n", num);

    free(sieve);
  }
  else
  {
    printf("Usage: pps <BASE> <MAX>\n");
  }
}
```

/EDIT

- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

03-01-2021, 04:03 AM **Post: #45**

robve
Member

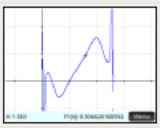
Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (02-28-2021 03:18 AM)
> [*]**robve** computed the integral using the **HP Prime** but alas, the result he posted is wrong. He also posted a *BASIC* program for the *SHARP PC-1350* which produced the *same* wrong result. Anyway, thanks for trying ...

Ah, that explains why I was getting nowhere with this one! Thanks for debugging my answer now that this challenge is over. Garbage in, garbage out: by putting the wrong expression in the integral the integration area did not offer any insights in what is weird about this integration. Somehow I missed the *LN* in the numerator of the integrand. The rest of the integrand expression was correct. Go figure...

With the *LN* correction applied to the numerator, I noticed that the Romberg integration ran much faster this time on the SHARP PC-1350 compared to the incorrect integrand. When viewing the value of `I` (the iterator) I noticed that `I=2` when the integration converges. Two successive trapezoidal approximations at alternating points are indistinguishable within *MachEps*. This means that the curve should be sufficiently close to linear between 1<=*x*<=*phi*. Plotting the integrand on the HP PRIME shows that this hunch is indeed the case:



Observing 0<=*f*(*x*)=(*x*-1)/(*phi*-1)<=1 for 1<=*x*<=*phi*. Integrating this f gives the same result 0.309016994375. Analytically, the value of the integral of *f* between 1 and *phi* is:

$$\frac{\phi^2-2\phi+1}{2\phi-2}=\frac{(1-\phi)^2}{2\phi-2}=-\frac{1}{2}\frac{(1-\phi)^2}{1-\phi}=\frac{\phi-1}{2}$$

Noticing this behavior of the quadrature algorithm is now (a bit late) helpful to answer what is weird about this integral. I am glad to use my trusty (albeit crusty) Romberg on the PC-1350.

Also, what is the fun of doing math and calc exercises if we don't implement numerical integration ourselves?

The Romberg quadrature program on the PC-1350 produces 0.3090169944 (10 digits exact) with the change `X=LN(X*X-V)` to line 300 to correct the integrand:

```
300 "F1" V=X,X=1.618033989,X=LN(X*X-V): GOSUB "GAMMA": W=Y
```

- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

03-01-2021, 08:05 PM **Post: #46**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

One more observation that may be of interest. Based on the visual clue that the integrant behaves as (x-1)/(phi-1), but is not the same, it is prudent to numerically verify that the function (x-1)/(phi-1) equals or closely approximates the integrand

$$\frac{\Gamma\ln(\phi^2-x)}{\Gamma\ln x+\Gamma\ln(\phi^2-x)}$$

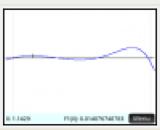
This can be done by listing the results of the two functions side-by-side for x between 1 and phi in increments of say 0.1 to compute the absolute or relative error of the difference between the two. We can also take the absolute difference of the two functions as the integrand and numerically evaluate the integral:

$$\int_1^\phi\left|\frac{\Gamma\ln(\phi^2-x)}{\Gamma\ln x+\Gamma\ln(\phi^2-x)}-\frac{x-1}{\phi-1}\right|dx\approx 5.66308603939\times 10^{-3}$$

The value is the result on the HP PRIME. My PC-1350 Romberg integrator gives the same 5.66308E-3. Therefore, the total area of the difference between the two functions is small, but not zero. Based on the numerical results, it does not appear convincing to state that

$$\frac{\Gamma\ln(\phi^2-x)}{\Gamma\ln x+\Gamma\ln(\phi^2-x)}\stackrel{?}{=}\frac{x-1}{\phi-1}$$

They are approximate, but not equal. Function (x-1)/(phi-1) under estimates the integrand on the interval 1<=x<=phi/2 and overestimates the integrand on interval phi/2<=x<=phi by about 0.014 symmetrically:



Because of this symmetry, one could argue that

$$\int_1^\phi\frac{\Gamma\ln(\phi^2-x)}{\Gamma\ln x+\Gamma\ln(\phi^2-x)}dx=\frac{\phi-1}{2}\approx 0.309016994375$$

Furthermore, the integral of the difference is near zero

$$\int_1^{\phi} \frac{\Gamma \ln(\phi^2 - x)}{\Gamma \ln x + \Gamma \ln(\phi^2 - x)} - \frac{x-1}{\phi - 1} dx \approx -1.13498070108 \times 10^{-13}$$

Suggesting that the over and under errors of the (x-1)/(phi-1) curve nicely cancel out over 1<=x<=phi.

- Rob

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

Valentin Albillo
Senior Member

Posts: 685
Joined: Feb 2015
Warning Level: 0%

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Hi all,

These are my *original solutions* for ***"Concoction the Fifth: Weird Primes"*** and ***"Concoction the Sixth: Weird Year"***, which concludes all my original solutions for the six *Concoctions* in this ***S&SMC #25***. You can follow these links to see my previously posted original solutions to:

"Concoction the First: Weird Limit"
"Concoction the Second: Weird Sum".
"Concoction the Third: Weird Integral" and "Concoction the Fourth: Weird Graph"

Note: My **HP-71B** code *might* use keywords from the *JPC ROM*, *MATH ROM*, *HP-IL ROM* and *STRINGLX LEX* file, executed on **go71b**, while *RPN* code is for the **HP-42S**, executed on a **DM42**.

My original solution for *"Concoction the Fifth: Weird Primes"*

"Consider a prime number so 'Perfectly Prime' (a PP for short) that changing any single [base 10] digit would turn it into a composite number. Write a program to compute: (a) the 5 smallest PP, (b) the first PP greater than 500 million, (c) the first PP greater than 777,777,777 and the second PP greater than 666,666,666."

What's so weird about these primes?

The Sleuthing

The first thing to do is to write a program which will find these *PP* starting from any given integer, and this straightforward *4-line* **HP-71B** program will nicely do:

```
1  DESTROY ALL @ INPUT K,N @ FOR P=1 TO K
2  N=FPRIM(N+2) @ N$=STR$(N) @ FOR I=1 TO LEN(N$) @ M$=N$ @ FOR D=0 TO 9
3  M$[I,I]=STR$(D) @ IF M$=N$ THEN 4 ELSE IF NOT PRIM(VAL(M$)) THEN 2
4  NEXT D @ NEXT I @ DISP P;N @ NEXT P
```

It asks how many *PP* to output and the *lower limit* to begin the search from. Let's run it to output the *5 smallest PP*, as asked. We begin at *11* as obviously no single-digit prime can be a *PP*:

```
[RUN]   ?  5,11 [ENDLINE]

        1    294001    { PP #1 }
        2    505447    { PP #2 }
        3    584141    { PP #3 }
        4    604171    { PP #4 }
        5    971767    { PP #5 }
```

As for the remaining questions, we proceed likewise:

```
[RUN]   ?  1,5E8        [ENDLINE]  ->    1   500004469     { PP #1318 }
```

```
[RUN]   ? 1,777777777 [ENDLINE]  ->   1  777781429    { PP #2259 }
[RUN]   ? 2,666666666 [ENDLINE]  ->   1  666850699    { PP #1845 }
                                      2  666999929    { PP #1846 }
```

The Results

Once the sleuthing's over, the results are as follows:

- The *5 smallest PP* are: ***294001***, ***505447***, ***584141***, ***604171*** and ***971767***.
- The *first PP > 500 million* is ***500004469***.
- The *first PP > 777,777,777* is ***777781429***.
- The *second PP > 666,666,666* is ***666999929***.

The Comments

Though for a prime being a *PP* seems a rare occurrence (the very first one is *294001*; that any do actually exist is *weird*), it's been proved that in fact there's an *infinity* of *PP* and what's more, a *positive* proportion (in terms of *asymptotic density*) of the primes are *PP*. That's ***weirder !***

We can try to (very roughly) "guesstimate" said proportion: there are *5,761,455* primes and *334 PP* less than *100 million*, so the rate comes out as ***0.00580 %***, i.e.: one *PP* per ~ *17,200 primes*. Going for the next order of magnitude, there are *50,847,534* primes and *3,167 PP* less than *1 billion*, so the rate now comes out as ***0.00623 %***, i.e.: one *PP* per ~ *16,000 primes*, hence the proportion, though very small, seems to be increasing (and in any case, it's asymptotically > *0* .)

Furthermore, if that wasn't weird enough, there's also a positive proportion of primes which are ***PPP*** *(**P**erfect **P**rime **P**lus)*, which have the property that if any single digit is changed *(including any zero among the prime's infinitely many <u>leading</u> zeros)*, then the resulting number is always *composite*. That's ***weirdest !!***

For instance, notice that the first *PP*, ***294001***, is <u>*not*</u> a ***PPP*** since changing, say, the *second* leading zero of *...0<u>0</u>0**294001*** results in *...0<u>1</u>0**294001***, which is prime. Matter of fact, even though a (much smaller, but still) positive proportion of the primes are ***PPP***, none are yet known.

Let's end these ***Comments*** by listing some peculiar *PP* you may find useful to check or optimize your code:

- The first five *PP* found above are *all* the *PP* less than 1 million. The next five are ***1062599***, ***1282529***, ***1524181***, ***2017963*** and ***2474431***.
- *PP #24* and *PP #25* are *consecutive* and *extremely close*: ***7469789*** and ***7469797***, respectively.
- *PP #51* and *PP #52* are *consecutive* and both look like nearby *date ranges*: ***1985_1991*** and ***2021_2327***, resp.
- *PP #1404* to *PP #1414* are *consecutive* and end in ***1***,***1***,***1***,***1***,***7***,***7***,***1***,***1***,***7***,***7***,***1***, resp.
- *PP #1846* and *PP #2539* have only *3* different digits each: ***666999929*** and ***844448333***, resp.
- *PP #3048* = ***969094909***, has the odd digit ***9*** in all *odd* positions and *even* digits in all *even* positions.

So much for *Perfect Primes* but there's more: you might want to try your hand at these four lingering questions (for which I *won't* provide answers, you'll be on your own):

- Are there any ***Twin PP***, i.e: two *PP* whose difference is ***2*** ?
- Are there any *PP* which remain so (i.e.: the result is always *composite*) when changing any ***two*** digits ?
- Find a ***PPP***. There's an infinity of them and you'll get worldwide fame in the math field if you do,
- What about ***Composites*** ? What's the first composite *(not divisible by 2 or 5)* so *"**P**erfectly **C**omposite"* [***PC***] that it remains *composite* when you change any single [base 10] digit ? (**hint**: it begins with ***2*** and ends with ***9***)

The Hall of Fame

This time the *one* expert which dealt with this *Concoction the Fifth: Weird Primes* is:

- **robve**, who wrote *HPPL* code for the **HP Prime** and found the correct answers to all four questions asked. He also wrote a *"Cheat program"* (his own words, mine too) in *C* which ran much faster *(D'oh!)*, but more importantly, he gave some theoretical considerations to estimate the chance that a prime number is a *PP* and correctly concluded that *"it seems reasonable to see perfect primes for large k [digits] and there are infinitely many of them"*.

  Afterwards he posted an alternative submission in *HPPL* and augmented his sleuthing above and beyond the call of duty, producing a list of the first *PP* for every base from *2* to *19*, then another list with the *PP* counts for the same bases, commenting on the results and the interesting pattern obtained. *That's* what sleuthing is all about !

  Finally he also said that *"It seems odd to me to change the leading digit to 0 to check for PP"* but the reason for that is made clearer in the light of the existence of *PPP*, as stated in the ***Comments*** above.

My original solution for *"Concoction the Sixth: Weird Year"*

"2020 shares a very striking numeric property with many other catastrophic years [...] try and discover what simple numeric property {which can be unambiguously stated by saying that the year's number "is a (five words)"} is shared by all the aforementioned numbers, and then write a program to output a listing of all years between AD 4 and AD 5000 (both included) which have this very property.

Additional questions are: ***(a)*** *How many years will be listed in the output ?* ***(b)*** *What will be the next predicted potentially catastrophic year after 2020 ? and* ***(c)*** *Should we be concerned ?*

The Sleuthing

Here, the very first thing to do is to discover the *"striking numeric property"* in question, which isn't as difficult as it seems because there are only so many such properties which can be unambiguously stated in just *five words*, e.g.: the number is a *"sum of some prime numbers"* does fit, but this is far too generic because every integer > *1* has that property. Many variations are possible (say *"5"* instead of *"some"* or replacing *"primes"* by *"squares"*, *"cubes"*, *"factorials"*, etc.)

One very useful heuristic is to analyze possible properties of the *smallest* numbers given: we first discover some property they have in common and then check if it applies to the bigger numbers as well. The numbers stated as sharing the sought-for property are ***458***, ***662***, ***666***, ***1348***, ***1556***, ***1849*** and ***2020***, so the smallest number is ***458*** and the sleuthing process begins with it.

After discarding many sums of *squares* and *cubes* (e.g.: *"sum of four square numbers"*, as *every* integer is a sum of four square numbers, including 0^2) and *"sum of some prime numbers"* and variations, we eventually discover that $\boldsymbol{458 = 13^2 + 17^2}$, a sum of exactly two *non-zero* squares, but regrettably ***662*** is *not*. We also notice that the numbers ***13*** and ***17*** are *primes* so we refine the property to *"sum of some squared primes"* but again that's *too generic* and gets us nowhere.

Cutting to the chase, eventually we also notice that **13** and ***17*** are *consecutive* primes so the property becomes *"sum of consecutive squared primes"* (5 words!) and this time we hit the jackpot: $\boldsymbol{458 = 13^2 + 17^2}$ and $\boldsymbol{662 = 3^2 + 5^2 + 7^2 + 11^2 + 13^2 + 17^2}$, and checking the bigger numbers we readily get:

> ***666*** $= 2^2 + 3^2 + 5^2 + 7^2 + 11^2 + 13^2 + 17^2$
> ***1348*** $= 13^2 + 17^2 + 19^2 + 23^2$
> ***1556*** $= 2^2 + 3^2 + 5^2 + 7^2 + 11^2 + 13^2 + 17^2 + 19^2 + 23^2$
> ***1849*** $= 43^2$ *{ a sum with just one summand is also a sum; the empty sum is 0 }*
> ***2020*** $= 17^2 + 19^2 + 23^2 + 29^2$

so ***bingo !***. Now it's just a matter of writing a program to find all the years in the given interval having that property, and this *7-line* program for the **HP-71B** will do:

```
1  DESTROY ALL @ OPTION BASE 1 @ DIM P(19) @ K=2 @ L=0 @ C=0
2  REPEAT @ L=L+1 @ P(L)=K*K @ K=FPRIM(K+1) @ UNTIL K>70
3  FOR N=4 TO 5000 @ J=L @ WHILE P(J)>N @ J=J-1 @ END WHILE
4  M=N @ I=J
5  M=M-P(I) @ IF NOT M THEN C=C+1 @ DISP N; @ GOTO 7
6  IF M<0 THEN J=J-1 @ GOTO 4 ELSE I=I-1 @ IF I THEN 5
7  NEXT N @ DISP @ DISP "Total:";C

[RUN]
             4      9     13     25     34     38     49
            74     83     87    121    169    170    195
```

204	208	289	290	339	361	364
373	377	**458**	529	579	628	650
653	**662**	**666**	819	841	890	940
961	989	1014	1023	1027	1179	**1348**
1369	1370	1469	1518	1543	1552	**1556**
1681	1731	1802	**1849**	**2020**	**2189**	2209
2310	2330	2331	2359	2384	2393	2397
2692	2809	2981	3050	3150	3171	3271
3320	3345	3354	3358	3481	3530	3700
3721	4011	4058	4061	4350	4489	4519
4640	4689	4714	4723	4727	4852	4899

Total: **91**

where the years given as examples appear in ***bold*** and the next one after *2020* appears in ***bold red***, which is ***2189*** = $\boldsymbol{13^2 + 17^2 + 19^2 + 23^2 + 29^2}$.

The Results

Based on the data obtained by the sleuthing process above and the results from running the program, the answers are:

- *The simple numeric property* is: The years' numbers are a *"**sum of consecutive squared primes**"* *(five words)*

- *The listing of all years between **AD 4** and **AD 5000*** (both included) which have this property is the output above, ***91*** years in all.

- *What will be the next predicted potentially catastrophic year after 2020 ?*: It will be ***2189***, as seen in the output.

- *Should we be concerned ?*: Well, with ever-advancing technology one can never say for sure, but I very much doubt any people reading this in ***2021*** will be kicking and alive by ***2189***, so ***no worries !***

The Comments

The joke explanation of why *2020* was such a *catastrophic* year is an original idea of mine, and implementing it was just a matter of finding some nice but simple numeric property of *2020*. After some sleuthing I found it to be that ***2020*** = $17^2 + 19^2 + 23^2 + 29^2$, which are the squares of consecutive primes (pretty nice indeed), and then I wrote the program to compute all other years between ***AD 4*** and ***AD 5000*** sharing this property. So far, so good.

Then I searched *Wikipedia* for a list of big catastrophes, and cross-checking the years found there with the years listed by my program I finally selected the ones most remarkable while ignoring numbers too low, to avoid reducing the difficulty too much (e.g.: if chosing *AD 13* it would be instantly recognizable as $2^2 + 3^2$, all too easy!) and *le voilà*, ***Concoction 6*** ready !

I also wrote the following *6-line* program for the **HP-71B** which accepts a given year in range and *demonstrates* whether it has the required numeric property (thus, if it indeed *was/might be catastrophic*) or not.).

```
1  DESTROY ALL @ OPTION BASE 1 @ DIM P(19),S$[80] @ K=2 @ L=0
2  REPEAT @ L=L+1 @ P(L)=K*K @ K=FPRIM(K+1) @ UNTIL K>70 @ INPUT N
3  J=L @ WHILE P(J)>N @ J=J-1 @ END WHILE
4  S$="" @ M=N @ I=J
5  M=M-P(I) @ S$=STR$(SQR(P(I)))&"^2+"&S$ @ IF NOT M THEN S$[LEN(S$)]="" @ DISP S$ @ END
6  IF M<0 THEN J=J-1 @ GOTO 4 ELSE I=I-1 @ IF I THEN 5 ELSE DISP "Not a sum"

[RUN]
      ?  2020    ->    17^2+19^2+23^2+29^2              { VAL(S$) = 2020 }
      ?  666     ->    2^2+3^2+5^2+7^2+11^2+13^2+17^2   { VAL(S$) = 666  }
      ?  1849    ->    43^2                             { VAL(S$) = 1849 }
      ?  555     ->    Not a sum
```

The Hall of Fame

Again, the *one* and *only* expert who dared to deal with this *Concoction the Sixth: Weird Year* is no other but

- **Gerson W. Barbosa**, who posted *RPL* code for such models as the **HP50G** and *BASIC* code for the **HP-75C**, both of which correctly produced the list of *"weird"* years, and he also gave their total number *(91)* and stated explicitly the remarkable property all these numbers share. Alas, he didn't post any details on his sleuthing, particularly how he managed to find the correct *"remarkable and striking numerical property"*, which would be fascinating for sure but never mind, ***Well Done !***

That's all, this concludes my *original solutions* for all ***6 Concoctions 6*** of this ***S&SMC #25*** of mine. **Thank you very much** to those who contributed their solutions or at least posted some comments, much appreciated.

I really hope you enjoyed it as well as all the readers in general. **Over and out**.

- ***Note**: For any comments not directly related to the math or code here but to ancillary matters such as this or that opinion on the rules or "Halls of Fame" or whatever, please **PM** me instead of posting them here. Let's keep this thread strictly mathematical and algorithmical in nature. Thanks.*

Best regards.
V.

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

PM WWW FIND EDIT QUOTE REPORT

03-02-2021, 10:46 AM (This post was last modified: 03-03-2021 10:14 AM by EdS2.) **Post: #48**

EdS2
Member

Posts: 274
Joined: Apr 2014

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Very nice, thanks again for the motivations and explanations.

Just one query: I'd interested to know the runtimes of some of these 71B programs. (Even if approximate, from memory.) For example, the time to find PP#1 and then PP#2. Another example, the most expensive calculation relating to the PP challenge - how long did that run for?

(If I had a 71B I could find this out directly!)

(Edit: answered in PM. VA (re)noting the use of go71b, about 128x faster than 71b, and taking 186s and 337s for PP's 1 and 2, and 4117s for PP# 1846. Noting also that a real 71b is about 10x or more the speed of a 41C. HP85 about 5x faster than 71. And noting that none of the code is written for speed.)

EMAIL PM FIND QUOTE REPORT

03-02-2021, 10:32 PM **Post: #49**

robve
Member

Posts: 51
Joined: Sep 2020

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

Very nice last challenge and results!

The combinatorial space of possible methods to apply seems quite large, perhaps too large for some to consider attacking this challenge when given limited time to work on this, despite helpful hints such as “five words”.

> **Valentin Albillo Wrote:** ⇒ (03-02-2021 03:55 AM)
>
> After discarding many sums of *squares* and *cubes* (e.g.: *"sum of four square numbers"*, as *every* integer is a sum of four square numbers, including 0^2)

I for sure could have guessed this challenge had something to do with a sum of squares for a specific year*.

> **Valentin Albillo Wrote:** ⇒ (03-02-2021 03:55 AM)
>
> Cutting to the chase, eventually we also notice that **13** and ***17*** are *consecutive* primes so the property becomes *"sum of consecutive squared primes"* (5 words!)

OK (duh), it happens that with many such methods other “special years” will coincidently fall in place too, with the often sought after number of the beast (which according to Numberphile is debatable whether it is 666 or 616 because both spell out Nero, the most evil ruler of the Roman era). If the solutions to the equation are very limited, then it won’t be likely. But if there are a couple of hundred solutions to an equation in a limited integer range and the definition of "special year" is less specific, then the odds are in favor of matching some special years.

The result is nice because of the series of primes is *consecutive*. Not only does this reduce the number of solutions reasonably, and thus the years that match, it also saves time not having to check all subsets of a certain set of integers (such as squares of primes). For example, checking all subsets of integers between 1 and *sqrt*(Y) for a given year Y is an $O(2^{Y/2})$ time process to compute. Reducing it to linear $O(Y)$ makes it not only fast, but much simpler to implement (on a calculator). The method is linear in Y to generate primes up to *sqrt*(Y) and to check if the sum of squares of consecutive primes 2 to *sqrt*(2) is equal to Y (i.e. in time $O(sqrt(Y^2))=O(Y)$).

Now that the results are in, I will share an effective approach to tackle the kinds of questions that require an exhaustive search among subsets. In general, we can approach these type of problems using the classic generate-and-test paradigm. Essentially brute force, but a bit smarter to generate combinations with a classic combination generating algorithm.

Listed below is a generic generate-and-test HPPL program to generate and check all subsets of a set of integers that sum up to a given integer, with a modification to sum up squares of primes to match a given year. First, the values are generated with `GENVALUES` and stored in list `L1`. Second, all subsets of the list are then tested to verify if their sum of squares match the given number `NUM`:

```
EXPORT GENVALUES(N)
BEGIN
LOCAL I;
  L1:=[2];
  FOR I FROM 3 TO N DO
    IF isprime(I) THEN
      L1:=append(L1,I);
    END;
  END;
END;

EXPORT SUMSQ(NUM)
BEGIN
  LOCAL N=FLOOR(SQRT(NUM)),K,I,F=0,X,Y;
  GENVALUES(N);
  FOR K FROM 1 TO SIZE(L1) DO
    L0:=MAKELIST(I,I,1,K);
    REPEAT
      L2:=MAKELIST(L1[L0[ I]],I,1,K);
      IF ΣLIST(L2^2)=NUM THEN
        PRINT(L2);
        F:=1;
      END;
      PNXCB(SIZE(L1),K,X,Y);
    UNTIL L0(K)=K;
    // IF F THEN
    //  RETURN;
    // END;
  END;
END;
```

Note that `L0:=MAKELIST(I,I,1,K)` is an index vector with values 1 to `K`. This vector is used to generate all subsets of values though indirect indexing, with elements of `L0` pointing to elements of `L1`. For example, `L0=[1,2,3]` is the subset of the first three primes `L1=[2,3,5,...]` whereas `L0=[1,4,7]` is the subset of three primes `L1=[2, , ,7, , ,17,…]`. The list `L2:=MAKELIST(L1[L0[ I]],I,1,K)` produces this list of subset of primes, which we then check if their sum of squares equals `NUM` with `ΣLIST(L2^2)=NUM`. If so, the list `L2` satisfies the conditions and is displayed.

Also note that we start with the smallest subsets first, a set of size 1, then a set of size 2, and so on by populating `L0:=MAKELIST(I,I,1,K)` for `K` from 1 to the size of `L1`.

The program displays the smallest subsets only. If all sets should be displayed, then uncomment the three lines in `SUMSQ`. To check with another function rather than squaring, just replace `L2^2` with another function and adjust `sqrt(NUM)` to assign to `N` accordingly (we limit the iteration space to `sqrt(NUM)` to avoid iterating over values who's square exceeds `NUM` and therefore are never part of the solution).

The workhorse for this program is the clever PNXCB combination generator which I've rewritten below in HPPL:

```
EXPORT PNXCB(N,K,X,Y)
BEGIN
  LOCAL J=1;
  WHILE 1 DO
    X:=L0(J);
    Y:=X+1;
```

```
    IF J=K THEN
      IF Y>N THEN
        Y:=K;
        L0(J):=Y;
        RETURN;
      END;
      L0(J):=Y;
      IF J>1 THEN
        L0(J-1):=X;
        X:=J-1;
      END;
      RETURN;
    END;
    IF Y<L0(J+1) THEN
      L0(J):=Y;
      IF J>1 THEN
        L0(J-1):=X;
        X:=J-1;
      END;
      RETURN;
    END;
    J:=J+1;
    X:=L0(J);
    Y:=X-1;
    IF Y>=J THEN
      L0(J):=Y;
      IF J>1 THEN
        Y:=J-1;
        L0(J-1):=Y;
      END;
      RETURN;
    END;
    IF J=K THEN
      Y:=N;
      L0(J):=Y;
      RETURN;
    END;
    J:=J+1;
  END;
END;
```

We start with an index list `L0` containing the values 1 to `K`. `PNXCB` takes `L0` as input. The next combination of the binomial *C*(*N*,*K*) (*N* choose *K*) is generated by `PNXCB` and is stored in `L0`.

`PNXCB` generates exactly *C*(*N*,*K*) selections *K* of *N* in `L0`. But we do not need to count them to stop the `REPEAT` loop! Here is the clever part: if we start with a list of values 1 to `K` then the `PNXCB` cycles through all combinations to arrive back to the initial list, exactly after *C*(*N*,*K*) calls to `PNXCB`. Therefore, the *C*(*N*,*K*) cycle is complete when the last value in the list is `K`, a useful property of `PNXCB`.

The `PNXCB` routine rewritten for classic BASIC calculators:

```
' Algorithm PNXCB Combination Generators
' PNXCB(P,N,K) cycles through "pointers" P(1:K) to N items
' Calling PNXCB NCR(N,K) times produces all combinations
' If initiallly P(I)=I, then NCR(N,K) cycle is complete when P(K)=K

' TEST PROGRAM
10 CLEAR: N=5,K=2: INPUT "N=";N,"K=";K
20 DIM P(K): FOR I=1 TO K: P(I)=I: NEXT I
30 FOR I=1 TO K: PRINT P(I);: NEXT I: PRINT
40 GOSUB 100
' all pointers were left-most initialized, if P(K)=K then end
50 IF P(K)=K END
60 GOTO 30

' PNXCB(P,N,K) -> P, X (old), Y (new), changes J
100 J=1: REM GOTO 180 FOR DOWN-UP SEQUENCE

110 X=P(J),Y=X+1
120 IF J<>K GOTO 160
130 IF Y>N LET Y=K,P(J)=Y: RETURN
```

```
140 P(J)=Y: IF J>1 LET P(J-1)=X,X=J-1
150 RETURN

160 IF Y<P(J+1) GOTO 140
170 J=J+1
180 X=P(J),Y=X-1
190 IF Y>=J GOTO 230
200 IF J=K LET Y=N,P(J)=Y: RETURN
210 J=J+1
220 GOTO 110

230 P(J)=Y: IF J>1 LET Y=J-1,P(J-1)=Y
240 RETURN
```

It's all basic, isn't it?

- Rob

*) the year 2021 has 17 minimal sets of 3 squares that sum up to 2021. Run `SUMSQ` with `GENVALUES` changed to `L1:=MAKELIST(I,I,1,N)` and uncomment the `IF F THEN` clause to display these minimal solutions.

HP Prime;Ti Nspire CXII CAS;Casio fx-CG50,fx-115ES+2;Sharp PC-G850VS,E500S,1475,1450,1360,1350,2500,1262,1500A

PM WWW FIND QUOTE REPORT

03-04-2021, 03:53 PM (This post was last modified: 03-04-2021 03:59 PM by Gerson W. Barbosa.) **Post: #50**

Gerson W. Barbosa
Senior Member

Posts: 1,369
Joined: Dec 2013

RE: [VA] Short & Sweet Math Challenge #25 "San Valentin's Special: Weird Math...

> **Valentin Albillo Wrote:** (03-02-2021 03:55 AM)
> **Gerson W. Barbosa**, who posted *RPL* code for such models as the **HP50G** and *BASIC* code for the **HP-75C**, ...
> ...
> Alas, he didn't post any details on his sleuthing, particularly how he managed to find the correct *"remarkable and striking numerical property"*, which would be fascinating for sure but never mind, ***Well Done !***

Hello, Valentín,

Thank you for the compliment and sorry for the belated reply (I've been busier than usual lately).

Unlike you, didactics is not one of my strengths, but I will try to demonstrate how I arrived at the solution to your sixth concoction. At first I was not intending to participate in any of these as I was 400 kilometers away from home and I had brought no calculator along. When I visited this thread again, all the other concoctions had been solved. It was late in the evening already, I had neither pencil nor paper but I found a copybook and a pen somewhere and decided to give it a go. As I am not smart enough to look at a small set of numbers and spot its common property right away, I submitted all your examples but one to Wolfram Alpha, which told me that
$2020 = 16^2 + 42^2 = 24^2 + 38^2$;
$1348 = 18^2 + 32^2$;
$1556 = 20^2 + 34^2$;
$458 = 13^2 + 17^2$;
$1849 = 43^2 + 0^2$ and
$666 = 15^2 + 21^2$.
"Quite easy!", I thought. These are just numbers that can be represented as a sum of two squares (although that didn't look like a 'striking' property to me). Then I wrote a simple RPN program on the emulated 49G+ on my smartphone running m48+:

```
« { } 0 71
FOR x x 71
FOR y x SQ y SQ + +
NEXT
NEXT SORT DUP SIZE 1 - 1 SWAP
FOR k k GETI UNROT GETI NIP ROT == { k { 0 } REPL } IFT
NEXT SORT
WHILE DUP HEAD NOT
REPEAT TAIL
END
»
```

After a rather long time, it returned a list with way more elements than the one specified by you (less than 100 elements), even when discarding the ones greater than 5000. Also, one of your examples were missing: 662. It turns out that was the only one I had left out.

I was disappointed of course, but not everything was lost. I noticed that two of your examples involved either the some of the squares of consecutive primes or the square of a prime: 458 and 1849. The Wikipedia article on 666 only confirmed that $666 = 2^2 + 3^2 + 5^2 + 7^2 + 11^2 + 13^2 + 17^2$. Likewise, it wasn't difficult to verify that all the remaining examples could be expressed as the sum of the squares of consecutive prime numbers. Now, I only needed to figure out a suitable algorithm. But at daft o'clock my reasoning was starting to fail, so I went to sleep and early next morning after I woke up I soon came out with a working program. The argument to the program in post #15 is the number of elements in the list of the squares of the first consecutive prime numbers, { 4, 9, 25, 49, 121, ... }. I started with a 15-element list, but I noticed the resulting list of 'weird years' would not include some of your examples. Then I increased it to 16, then to 20, when the list seemed to be complete (the list up to 5000 would not be changed by further increases). I thought it would be nice if the program calculated the size of the basic list automatically, so I researched at OEIS, which led me to an interesting recent article on the subject. That's where I 'borrowed' the formula I used in my next programs, which has to do with the size of the list of the first squares of primes necessary to produce the list of all years up to certain limit (hopefully I have simplified the correct formula – the article looks complicated to me). Next day I was tired after driving all the way back home, but I took the time to write an HP-75C version per your suggestion of using programming languages other than RPL for more clarity. But it payed off, as I noticed the BASIC program was more optimized than the RPL program (the output list never would grow longer than it needed to be). As a result, I was able to write a faster RPL program, but really expert RPL programmers, which I am not, can optimized it even more.

Thank you also for taking the time to concoct these S&SMCs. Even when we don't participate it's a joy just to read them, especially your final comments and solutions.

Best regards,

Gerson.

User(s) browsing this thread: Valentin Albillo*