```
> ### Example 7.11. p = 55681 for (2.5).mws
```

A role for generalised Fermat numbers

by

John B. Cosgrave and Karl Dilcher

A reminder of the meaning of our paper's congruence (2.5): for $n = p^{\alpha}\, q_1^{\beta_1}\, q_2^{\beta_2} \ldots q_s^{\beta_s}$, distinct primes $p, q_1, q_2, \ldots, q_s = 1, -1, -1, \ldots, -1 \pmod 3$ we seek solutions of the Gauss factorial congruence

$$\mathrm{floor}\left(\left(\frac{1}{3}\{n-1\}\right)_n!\right) = 1 \pmod n$$

In this Maple-to-pdf conversion we exhibit all solutions of our paper's congruence (2.5) for the prime $p = 55681$, a specially chosen non-standard Jacobi prime.

It should be emphasised that for $p = 55681$ the '$s = 8$' is the current limit for which we can make a definitive statement concerning the exact number of solutions of (2.5). A similar statement for $s = 9$ would require knowing the complete factorisation of the 608-digit $55681^{2^7} + 1$ (that '7' being $9 - 2$).

This $p = 55681$ satisfies $ord_p\left(\frac{1}{3}\{p-1\}\right) = 3.2^6$, and $ord_p\left(\frac{1}{6}\{p-1\}\right) = 3.2^5$.

Procedures

```
> with(numtheory): ### needed for 'order'
   with(combinat): ### needed for 'choose'
  ### 01:
  PI := proc(n, M, i) local k, r; r := 1:
        for k from floor(((i-1)*(n-1)/M + 1)) to i*(n-1)/M do
          if igcd(n, k) = 1 then r := mods(r*k, n); fi; od; r; end:
  ### 02:
  PRFAC := proc(le, la, p, alpha) local r, k, MOD; r := le; MOD := p^alpha;
           for k from (le+1) to la do r := mods(r*k, MOD) od; r; end:
  ### 03:
  the_ones := proc(n, M) local L, p; L := []: for p in factorset(n) do if p mod M = 1 then
              L := [op(L), p] fi od; L; end:
  ### 04:
  the_minus_ones := proc(n, M) local L, p; L := []: for p in factorset(n) do if mods(p, M) = -1
                    then L := [op(L), p] fi od; L; end:
  ### 05:
```

```
Pow := proc(n, p) local t, a; t := n: a := 0: while t mod p = 0 do t := t/p: a := a+1: od: a;
end:

### 06:

### "more" simply means that 's' (in application) > 1
### (i.e., there is 'more' than ONE 'q')

### Bear in mind that in using this procedure we are making the understanding that p
### occurs only to the 1st power, and 'w' will be square-free ('primitive' solutions)

### 'L' REPLACES the INITIAL 'w'
### 'w' becomes a local variable, DEFINED as the product of the members of L (the 'q')

PI3_more_modified := proc(L, s, p, fac) local w, EF, FAC, R;

        if s < 2 then lprint(`Remember that s should be greater than 1.`); RETURN(); fi;

            w := mul(q, q in L):

           EF := mods(w&^((p - 1)/3), p):      ### Euler-Fermat element for s > 1.

          FAC := mods(fac&^(2^s), p):          ### This is the Gauss 3 closed form mod p

     if mods(w, 3) = 1 and s mod 2 = 0 then

             R := mods(FAC / EF, p):

   elif mods(w, 3) = 1 and s mod 2 = 1 then

             R := mods(EF * FAC, p):

   elif mods(w, 3) = -1 and s mod 2 = 0 then

             R := mods(EF / FAC, p):

   elif mods(w, 3) = -1 and s mod 2 = 1 then

             R := mods( 1 / (EF * FAC), p):

      fi;

    R; end:

#####################################################################

### The following PI3 - of which we make only limited use - tests some individual n-values
### It allows for having 'alpha' > 1

### The following PHI3 uses the D.H. Lehmer formulae for the
### PHI-values of w = 1/-1 (mod 3) for w having NO prime factor = 1 (mod 3)

### 07:

PHI3 := proc(w) local s; s := nops(factorset(w)):

   if w mod 3 = 1 then (phi(w) + 2^(s-1))/3 elif mods(w, 3) = -1 then (phi(w) - 2^(s-1))/3 fi;
end:

### 08:

PI3 := proc(n) local p, a, Gf, w, s, signs, PHIw3, Sw, Q, EF, Rpa, Rw, R;

            p := op(the_ones(n, 3)):                    ### This gives 'p'
            a := Pow(n, p):                             ### This gives 'a', i.e. 'alpha'

  if a = 1 then Gf := PRFAC(1, (p-1)/3, p, 1) elif a > 1 then Gf := PI(p^a, 3, 1) fi:


      ### This is ((p^a - 1)/3)_p! mod p^a, speeded up in the case a(lpha) = 1

            w := n/(p^a):                          ### This is 'w'
            s := nops(factorset(w)):               ### This is 's'
        signs := (-1)^(s-1):                       ### This is '1' at ODD 's', and '-1' at EVEN 's'
        PHIw3 := PHI3(w):                         ### This is the FASTER improvement on PHI(3, 1,
w)

           Sw := factorset(w):                    ### This is the set of all ('s' of) the 'q'
            Q := mul(q, q = Sw):                  ### This is the product of all ('s' of) the 'q'
           EF := mods(Q&^(phi(p^a)/3), p^a):      ### the Euler-Fermat element.

          Rpa := mods(EF^signs * Gf&^(2^s), p^a): ### Note the SIGN element in the EF term

           Rw := mods(1/p&^PHIw3, w):               ### There is NO SIGN element here at Gauss 3
```

```
    if w = 2 then
             R := mods(chrem([-1/Rpa, Rw], [p^a, w]), n): ### Note the '-'
  elif w <> 2 and mods(w, 3) = -1 then
             R := mods(chrem([1/Rpa, Rw], [p^a, w]), n):  ### Note the 1/Rpa
  elif mods(w, 3) = 1 then
             R := mods(chrem([Rpa, Rw], [p^a, w]), n):
    fi; R; end:
```

Gauss 3 support primes for *p* = 55681 (one Gauss-3 level-8 support has 249 digits)

```
>               p := 55681:
     level||1[p] := [2, 5, 29]:                          ### These divide (55681 - 1)
     level||2[p] := [11, 2531]:                          ### These divide (55681 + 1)
     level||3[p] := []:                                  ### No 'q' for (55681^2 + 1)
     level||4[p] := [41, 121477457]:                     ### These divide (55681^(2^2) + 1)
     level||5[p] := [17, 12075324422351249]:             ### These divide (55681^(2^3) + 1)
     level||6[p] := []:                                  ### No 'q' dividing 556811^(2^4) + 1
                                                         ### Our COMPLETE factorisation
     level||7[p] := [257, 610817, 4766007046199118910 73, 49403957554237215440934 6497]:
```

```
     level||7[p] := [257, 610817, 476600704619911891073, 494039575542372154409346497]:

                                                         ### These divide 55681^(2^5) + 1
                                                         ### Our COMPLETE factorisation
                                                         ###  29th Oct 2013

     level||8[p] := [652980135409104837678582610371183252063988 49,

  5430065006389584982184906089683575088937603970131954895946359874894517106136643531669045869571635
  9316934391090865493136004628257534364175802467156932863636609772154081500828008188169170737952634
  967989897592460040649822049644112410252192520231933 7217]:
                                                     ### These divide 55681^(2^6) + 1
                                                     ### Our COMPLETE factorisation
                                                     ### Tues 27th Jan 2015

                              ### COMPLETE TO HERE

          LEV := 8:
  LEVEL||LEV[p] := []:
    for lev to LEV do
  for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
> print(``); LEVEL||LEV[p]; print(``);
```

[2, 5, 29, 11, 2531, 41, 121477457, 17, 12075324422351249, 257, 610817, 476600704619911891073, 494039575542372154409346497,
652980135409104837678582610371183252063988 49,
5430065006389584982184906089683575088937603970131954895946359874894517106136643531669045869571635931693 4391\
09086549313600462825753436417580246715693286363660977215408150082800818816917073795263496798989759246004064\
98220496441124102521925202319337217]

(2.1)

```
> seq(isprime(q), q = LEVEL||LEV[p]);
```

true, true, true, true, true, true, true, true, true, true, true, true, true, true, true

(2.2)

```
> nops(LEVEL||LEV[p]);
```
$$15 \quad (2.3)$$

```
> seq(q mod 3, q = LEVEL||LEV[p]);
```
$$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 \quad (2.4)$$

```
> print(``);
  if level||1[p] <> [] then print([[1], seq(p - 1 mod q, q = level||1[p])]); fi:
  if level||2[p] <> [] then print([[2], seq(p + 1 mod q, q = level||2[p])]); fi:
  for LEV from 3 to 8 do if level||LEV[p] <> [] then
  print([[LEV], seq(p&^(2^(LEV - 2)) + 1 mod q, q = level||LEV[p])]); fi od;
```
[[1], 0, 0, 0]
[[2], 0, 0]
[[4], 0, 0]
[[5], 0, 0]
[[7], 0, 0, 0, 0]
[[8], 0, 0] (2.5)

```
>             p := 55681;
          fac := PRFAC(1, (p-1)/3, p, 1);  ### Note the '3'
```
$p := 55681$

$fac := -12001$ (1)

__________[start of s = 6]

```
> Pow(55681^(2^5) - 1, 2); ### This '12' is 1 (from the primitive solutions) + (the extra) '11' (= 7+1+1+1+1)
```
12 (2)

```
> p := 55681: ifactor(p - 1);
```
$$(2)^7 (3) (5) (29) \quad (3)$$

This '12' is the highest power of 2 dividing the product

$$(\mathbf{55681} - \mathbf{1})\ (55681 + 1)\ (55681^2 + 1)\left(55681^{2^2} + 1\right) \ldots \left(55681^{2^{4 (= 6 - 2)}} + 1\right)$$

By Theory, since $p = 55681$ is Gauss 3 level 6, then the least possible solution (with $2 \le s$) is at s

= 6.

$s = 6$, BOUND = 9. 33 primitive solutions, the least the 18-digit

$$n = 366810976138857910 = 2 * 5 * 11 * 17 * 29 * 121477457 * 55681,$$

the largest had 36 digits. The 21 even solutions each generate (by theory) an extra 11 solutions

The largest even solution is the 39-digit: 2^{12} . 17 . 41 . 2531 . 121477457 . 12075324422351249 . 55681

```
> length(2^11 * mul(q, q = [2, 17, 41, 2531, 121477457, 12075324422351249]) * 55681);
```

39 **(3.1)**

```
>              p := 55681:
             fac := -12001:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
             LEV := 6:
  LEVEL||LEV[p] := []:
   for lev to LEV do
  for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
  print(``); S_potential||p := LEVEL||LEV[p];
           count := 0:
          SOLN_L := []:
  print(``); print(____________________); print(``);
  for L_ in choose(S_potential||p, LEV) do if PI3_more_modified(L_, LEV, p, fac) = 1 then
           count := count+1:
  print(``); print(L_); print(``); lprint(`Here is a solution:`); print(``);
  print(p*mul(j, j = L_));
          SOLN_L := [op(SOLN_L), p*mul(j, j = L_)]:
  print(``); lprint(`which has`, length(p*mul(j, j = L_)), `digits.`); print(____________________);
  fi; od:
  print(``); lprint(`There were`, count, `solutions altogether.`); print(____________________);
  print(``);
```

S_potential55681 := [2, 5, 29, 11, 2531, 41, 121477457, 17, 12075324422351249]

____

[2, 5, 11, 17, 29, 121477457]

`Here is a solution:`

366810976138857910

`which has`, 18, `digits.`

[2, 5, 11, 17, 2531, 121477457]

`Here is a solution:`

32013744158877564490

`which has`, 20, `digits.`

[2, 5, 11, 17, 121477457, 12075324422351249]

`Here is a solution:`

152736604777794902495471576621710

`which has`, 33, `digits.`

[2, 5, 11, 29, 41, 121477457]

`Here is a solution:`

884661765981951430

`which has`, 18, `digits.`

[2, 5, 11, 29, 2531, 12075324422351249]

`Here is a solution:`

5428610247310121013435593410

`which has`, 28, `digits.`

[2, 5, 11, 41, 2531, 121477457]

`Here is a solution:`

77209618265528243770

`which has`, 20, `digits.`

[2, 5, 11, 41, 121477457, 12075324422351249]

`Here is a solution:`

368364752699387706018490273028830

`which has`, 33, `digits.`

[2, 5, 17, 29, 2531, 121477457]

`Here is a solution:`

84399870964313579110

`which has`, 20, `digits.`

[2, 5, 17, 29, 121477457, 12075324422351249]

`Here is a solution:`

402669230777822924760788702002690

`which has`, 33, `digits.`

[2, 5, 17, 2531, 121477457, 12075324422351249]

`Here is a solution:`

35143304244781718019639869129958910

`which has`, 35, `digits.`

[2, 5, 29, 41, 2531, 121477457]

`Here is a solution:`

203552629972756279030

`which has`, 21, `digits.`

[2, 5, 29, 41, 121477457, 12075324422351249]

`Here is a solution:`

971143438934749406776019810712370

`which has`, 33, `digits.`

[2, 5, 41, 2531, 121477457, 12075324422351249]

`Here is a solution:`

84757380825650025812072625548724430

`which has`, 35, `digits.`

[2, 11, 17, 29, 41, 121477457]

`Here is a solution:`

3007850004338634862

`which has`, 19, `digits.`

[2, 11, 17, 29, 2531, 12075324422351249]

`Here is a solution:`

18457274840854411445681017594

`which has`, 29, `digits.`

[2, 11, 17, 41, 2531, 121477457]

`Here is a solution:`

262512702102796028818

`which has`, 21, `digits.`

[2, 11, 17, 41, 121477457, 12075324422351249]

`Here is a solution:`

1252440159177918200462866928298022

`which has`, 34, `digits.`

[2, 11, 29, 41, 2531, 12075324422351249]

`Here is a solution:`

44514604027942992310171865962

`which has`, 29, `digits.`

[2, 17, 29, 41, 2531, 121477457]

`Here is a solution:`

692078941907371348702

`which has`, 21, `digits.`

[2, 17, 29, 41, 121477457, 12075324422351249]

`Here is a solution:`

3301887692378147983038467356422058

`which has`, 34, `digits.`

[2, 17, 41, 2531, 121477457, 12075324422351249]

`Here is a solution:`

288175094807210087761046926865663062

`which has`, 36, `digits.`

[5, 11, 17, 29, 2531, 121477457]

`Here is a solution:`

464199290303724685105

`which has`, 21, `digits.`

[5, 11, 17, 29, 121477457, 12075324422351249]

`Here is a solution:`

2214680769278026086184337861014795

`which has`, 34, `digits.`

[5, 11, 17, 2531, 121477457, 12075324422351249]

`Here is a solution:`

193288173346299449108019280214774005

`which has`, 36, `digits.`

[5, 11, 29, 41, 2531, 121477457]

`Here is a solution:`

1119539464850159534665

`which has`, 22, `digits.`

[5, 11, 29, 41, 121477457, 12075324422351249]

`Here is a solution:`

5341288914141121737268108958918035

`which has`, 34, `digits.`

[5, 11, 41, 2531, 121477457, 12075324422351249]

`Here is a solution:`

466165594541075141966399440517984365

`which has`, 36, `digits.`

[5, 17, 29, 2531, 121477457, 12075324422351249]

`Here is a solution:`

509577911549334911284778102384404195

`which has`, 36, `digits.`

[5, 29, 41, 2531, 121477457, 12075324422351249]

`Here is a solution:`

1228982021971925374275053070456504235

`which has`, 37, `digits.`

[11, 17, 29, 41, 2531, 121477457]

`Here is a solution:`

3806434180490542417861

`which has`, 22, `digits.`

[11, 17, 29, 41, 121477457, 12075324422351249]

`Here is a solution:`

18160382308079813906711570460321319

`which has`, 35, `digits.`

[11, 17, 41, 2531, 121477457, 12075324422351249]

`Here is a solution:`

1584963021439655482685758097761146841

`which has`, 37, `digits.`

[17, 29, 41, 2531, 121477457, 12075324422351249]

`Here is a solution:`

4178538874704546272535180439552114399

`which has`, 37, `digits.`

`There were`, 33, `solutions altogether.`

(3.2)

```
> sort([seq(length(n), n = SOLN_L)]);
```

[18, 18, 19, 20, 20, 20, 21, 21, 21, 21, 22, 22, 28, 29, 29, 33, 33, 33, 33, 34, 34, 34, 34, 35, 35, 35, 36, 36, 36, 36, 37, 37, 37] **(3.3)**

```
> print(``); L2 := []: for n in SOLN_L do if n mod 2 = 0 then L2 := [op(L2), n] fi od; L2; print
  (``);
```

[366810976138857910, 320137441588775644490, 15273660477779490249547157662l710, 884661765981951430,
542861024731012101343559341O, 772096182655282437770, 368364752699387706018490273028830, 843998709643l3579110,
402669230777822924760788702002690, 351433042447817180196398691299589l0, 203552629972756279030,
971143438934749406776019810712370, 847573808256500258120726255487244 30, 3007850004338634862,
1845727484085441144568101 7594, 262512702102796028818, 125244015917791820046286692829 8022,
4451460402794299231017186 5962, 692078941907371348702, 330188769237814798303846735642 2058,
288175094807210087761046926865663062] **(3.4)**

```
> [nops(L2)];

  seq(length(n), n = sort(L2));
```

[21]

18, 18, 19, 20, 20, 20, 21, 21, 21, 28, 29, 29, 33, 33, 33, 33, 34, 34, 35, 35, 36 **(3.5)**

```
> L2sort := sort(L2):
```

```
> for n in L2sort do print(``); ifactor(n); seq(PI3(2^j*n), j = 1..12); od;
```

(2) (5) (11) (17) (29) (121477457) (55681)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -7512288791323809996 79

(2) (5) (11) (29) (41) (121477457) (55681)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -181178729673103652863 9

(2) (11) (17) (29) (41) (121477457) (55681)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -616007680888552419737 5

(2) (5) (11) (17) (121477457) (55681) (2531)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -655641480373812520755 19

(2) (5) (11) (41) (121477457) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1581252982078018432409559

(2) (5) (17) (29) (121477457) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1728509357349142100172799

(2) (5) (29) (41) (121477457) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -4168757861842048594534439

(2) (11) (17) (41) (121477457) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -5376260139065262670192663

(2) (17) (29) (41) (121477457) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -14173776730262965221416955

(2) (5) (11) (29) (12075324422351249) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -111177937864911278355160953036799

(2) (11) (17) (29) (12075324422351249) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -378004988740698346407547240325111

(2) (11) (29) (41) (12075324422351249) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -911659090492272482512319814901755

(2) (5) (11) (17) (121477457) (12075324422351249) (55681)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -3128045665849239603107257889212620799

(2) (5) (11) (41) (121477457) (12075324422351249) (55681)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -7544110135283460219258680791630438399

(2) (5) (17) (29) (121477457) (12075324422351249) (55681)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -8246665846329813499100952617015091199

(2) (5) (29) (41) (121477457) (12075324422351249) (55681)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -19889017629383667850772885723389337599

(2) (11) (17) (41) (121477457) (12075324422351249) (55681)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -25649974459963764745479514691543490555

(2) (17) (29) (41) (121477457) (12075324422351249) (55681)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -67622659939904470692627811459523747833

(2) (5) (17) (121477457) (12075324422351249) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -71973487093312958504222451978155847679

(2) (5) (41) (121477457) (12075324422351249) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -173583115930931252863124737123787632639

(2) (17) (41) (121477457) (12075324422351249) (55681) (2531)
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -590182594165166259734624106220877950975 **(3.6)**

The total number of solutions at $s = 6$ of floor$\left(\left(\frac{1}{3}\ \{n-1\}\right)_n!\right) = 1 \pmod{n}$ is 276:

```
> no_solns_6 := (33-21) ### the number of ODD primitive solutions
              + 21 * (1 + 11);
```

$$no\_solns\_6 := 264 \tag{4}$$

__________[end of s = 6, start of s = 7]

```
> Pow(55681^(2^6) - 1, 2); ### This '13' is 1 (from the primitive solutions)
                          ### + (the extra) '12' (= 7+1+1+1+1+1)
```

$$13 \tag{5}$$

This '13' is the highest power of 2 dividing the product

$$(55681 - 1)\ (55681 + 1)\ \left(55681^2 + 1\right)\left(55681^{2^2} + 1\right)\ldots\left(55681^{2^{5(=7-2)}} + 1\right)$$

$s = 7$, BOUND = 13. 576 primitive solutions (572 predicted), the least the 15-digit

$n = 313344901555730 = 2 * 5 * 11 * 17 * 29 * 41 * 2531 * 55681$,

the largest had 88 digits. The largest even solution (see details in next section) has 89 digits

```
> N := binomial(13, 7): floor(N/3);
```

$$572 \tag{4.1}$$

```
>              p := 55681:
             fac := -12001:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
             LEV := 7:
  LEVEL||LEV[p] := []:
   for lev to LEV do
 for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
 print(``);
 S_potential||p := LEVEL||LEV[p];
           count := 0:
             MIN := 10^100: ### Clearly greater than ANY small solution
             MAX := 0:
           print(``);
  for L_ in choose(S_potential||p, LEV) do if PI3_more_modified(L_, LEV, p, fac) = 1 then
              w := mul(q, q = L_):
```

```
            MIN := min(MIN, p*w):
            MAX := max(MAX, p*w):
          count := count+1:
  fi; od:
            lprint(`There were`, count, `solutions altogether.`);
  print(``); lprint(`The smallest solution is:`); print(``); MIN;
  print(``); lprint(`The largest solution is:`); print(``); MAX; print(``);
```

S_potential55681 := [2, 5, 29, 11, 2531, 41, 121477457, 17, 12075324422351249, 257, 610817, 47660070461991891073, 4940395755423721544093464 97]

`There were`, 576, `solutions altogether.`

`The smallest solution is:`

313344901555730

`The largest solution is:`

7641072668218240002550711786912664636241990305932316995229655352833267820425563452190147

(4.2)

```
> length(MIN); length(MAX); ifactor(MIN);
```

15

88

(2) (5) (11) (17) (29) (41) (55681) (2531) **(4.3)**

```
> n := 313344901555730: ifactor(n); seq(PI3(2^j*n), j = 1..13);
```

(2) (5) (11) (17) (29) (41) (55681) (2531)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1283460716772270079 **(4.4)**

This extra $s = 7$ section is to deal with the even solutions:

even solutions at $s = 7$, BOUND = 13. 312 primitive solutions, the least the (same) 15-digit

$$n = 313344901555730 = 2 * 5 * 11 * 17 * 29 * 41 * 2531 * 55681,$$

the largest (of the even primitives, that is) had 85 digits

```
>              p := 55681:
            fac := -12001:   ### THE ABOVE PRE-COMPUTED VALUE OF fac
            LEV := 7:
   LEVEL||LEV[p] := []:
    for lev to LEV do
  for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
```

```
print(``);
S_potential||p := LEVEL||LEV[p];
         count := 0:
           MIN := 10^100: ### Clearly greater than ANY small solution
           MAX := 0:
         print(``);
 for L_ in choose(S_potential||p, LEV) do if
     member(2, L_) and PI3_more_modified(L_, LEV, p, fac) = 1 then
             w := mul(q, q = L_):
           MIN := min(MIN, p*w):
           MAX := max(MAX, p*w):
         count := count+1:
fi; od:
           lprint(`There were`, count, `solutions having q =  2 as support.`);
print(``); lprint(`The smallest solution is:`); print(``); MIN;
print(``); lprint(`The largest solution is:`); print(``); MAX; print(``);
```

S_potential55681 := [2, 5, 29, 11, 2531, 41, 121477457, 17, 12075324422351249, 257, 610817, 476600704619911891073, 4940395755423721544409346497]

`` `There were`, 312, `solutions having q =  2 as support.` ``

`` `The smallest solution is:` ``

313344901555730

`` `The largest solution is:` ``

603798709460153299292825901771052124554878728244355353238218518596070155703323860307 (5.1)

```
> length(MIN); length(MAX); ifactor(MIN);
```

15

85

(2) (5) (11) (17) (29) (41) (55681) (2531) (5.2)

And the largest even solution has this number of digits:

```
> length(2^12 * MAX); ### the LARGEST EVEN solution has this number of digits:
```

89 (5.3)

```
> n := 313344901555730: ifactor(n); seq(PI3(2^j*n), j = 1..13);
```

(2) (5) (11) (17) (29) (41) (55681) (2531)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1283460716772270079 (5.4)

The total number of solutions at $s = 7$ of

$$\text{floor}\left(\left(\frac{1}{3}\{n-1\}\right)_n!\right) = 1 \pmod{n}$$ is 4320**:**

```
> no_solns_7 := (576-312) ### the number of ODD primitive solutions
               + 312 * (1 + 12);
```

$$no\_solns\_7 := 4320 \tag{6}$$

__________[end of s = 7, start of s = 8]

```
> Pow(55681^(2^7) - 1, 2); ### This '14' is 1 (from the primitive solutions)
                         ### + (the extra) '13' (= 7+1+1+1+1+1+1)
```

$$14 \tag{7}$$

This '14' is the highest power of 2 dividing the product

$$(55681-1)\ (55681+1)\ (55681^2+1)\left(55681^{2^2}+1\right)\ldots\left(55681^{2^{6(=8-2)}}+1\right)$$

s **= 8, BOUND = 15. 2151 primitive solutions (2145 predicted), the least the 17-digit**

$$n = 80529639699822610 = 2 * 5 * 11 * 17 * 29 * 41 * 257 * 2531 * 55681,$$

the largest had 379 digits. By coincidence the largest even solution (see details in next section) also has 379 digits

```
> N := binomial(15, 8): floor(N/3);
  2531*(257**55681)
```

$$2145 \tag{6.1}$$

```
>              p := 55681:
             fac := -12001:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
             LEV := 8:
  LEVEL||LEV[p] := []:
   for lev to LEV do
  for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
  print(``);
  S_potential||p := LEVEL||LEV[p];
```

```
          count := 0:
            MIN := 10^100: ### Clearly greater than ANY small solution
            MAX := 0:
          print(``);
 for L_ in choose(S_potential||p, LEV) do if PI3_more_modified(L_, LEV, p, fac) = 1 then
              w := mul(q, q = L_):
            MIN := min(MIN, p*w):
            MAX := max(MAX, p*w):
          count := count+1:
 fi; od:
            lprint(`There were`, count, `solutions altogether.`);
 print(``); lprint(`The smallest solution is:`); print(``); MIN;
 print(``); lprint(`The largest solution is:`); print(``); MAX; print(``);
```

S_potential55681 := [2, 5, 29, 11, 2531, 41, 121477457, 17, 12075324422351249, 257, 610817, 476600704619911891073, 494039575542372154409346497, 65298013540910483767858261037118325206398849, 54300650063895849821849060896835750889376039701319548959463598748945171061366435316690458695716359316934391\
09086549313600462825753436417580246715693286363660977215408150082800818816917073795263496798989759246004064\
98220496441124102521925202319337217]

`There were`, 2151, `solutions altogether.`

`The smallest solution is:`

80529639699822610

`The largest solution is:`

1054207751025532246936169975003755885073912404533718638929237714585186118111423661111196530574456060451789216082\
10048393957257884579114074839460209131487681326682808838649619146307949602852263635977395417778692268870449\
60958257779598485693525564987343986764904571852183005349313452420778744517218520118675171788022742566312101\
90249789394441257230111952557358965417814346147800643

(6.2)

```
> length(MIN); length(MAX); ifactor(MIN);
```

17

379

(2) (5) (11) (17) (29) (41) (257) (55681) (2531)

(6.3)

```
> n := 80529639699822610: ifactor(n); seq(PI3(2^j*n), j = 1..14);
```

(2) (5) (11) (17) (29) (41) (257) (55681) (2531)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, −659698808420946821119

(6.4)

This extra $s = 8$ section is to deal with the **even** solutions:

even solutions at $s = 8$, BOUND = 13. 1152 primitive solutions, the least the (same) 17-digit

$$n = 313344901555730 = 2 * 5 * 11 * 17 * 29 * 41 * 257 * 2531 * 55681,$$

the largest (of the even primitives, that is) had 379 digits

```
>              p := 55681:
             fac := -12001:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
             LEV := 8:
   LEVEL||LEV[p] := []:
    for lev to LEV do
  for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
  print(``);
  S_potential||p := LEVEL||LEV[p];
           count := 0:
             MIN := 10^100: ### Clearly greater than ANY small solution
             MAX := 0:
           print(``);
   for L_ in choose(S_potential||p, LEV) do if
       member(2, L_) and PI3_more_modified(L_, LEV, p, fac) = 1 then
               w := mul(q, q = L_):
             MIN := min(MIN, p*w):
             MAX := max(MAX, p*w):
           count := count+1:
  fi; od:
             lprint(`There were`, count, `solutions having q =  2 as support.`);
  print(``); lprint(`The smallest solution is:`); print(``); MIN;
  print(``); lprint(`The largest solution is:`); print(``); MAX; print(``);
```

S_potential55681 := [2, 5, 29, 11, 2531, 41, 121477457, 17, 12075324422351249, 257, 610817, 476600704619911891073,
494039575542372154409346497, 65298013540910483767858261037118325206398849,
54300650063895849821849060896835750889376039701319548959463598748945171061366435316690458695716359316934391\
09086549313600462825753436417580246715693286363660977215408150082800818816917073795263496798989759246004064\
98220496441124102521925202319337217]

`There were`, 1152, `solutions having q =  2 as support.`

`The smallest solution is:`

80529639699822610

`The largest solution is:`

8330365476298160781795100553170730028241109478733454278381965346386298839284264410203054370402655554735592383106\
28592603376198218720774988853893394954466071328983080510862261132421569362720376420208576987583502717269455\
62688722082959191572703002665697248241047584766361164356487178354632512976835401965034941035343678911988162\
0110461789364881256508852277644381997482691543106 **(7.1)**

```
> length(MIN); length(MAX); ifactor(MIN);
```

17

375

(2) (5) (11) (17) (29) (41) (257) (55681) (2531) **(7.2)**

And the largest even solution has this number of digits:

```
> length(2^13 * MAX);
```

$$379 \tag{7.3}$$

```
> n := 80529639699822610: ifactor(n); seq(PI3(2^j*n), j = 1..14);
```

$$(2)\,(5)\,(11)\,(17)\,(29)\,(41)\,(257)\,(55681)\,(2531)$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -6596988084209468211119 \tag{7.4}$$

The total number of solutions at $s = 8$ of $\mathrm{floor}\left(\left(\frac{1}{3}\ \{n-1\}\right)_n !\right) = 1 \pmod{n}$ is 17127:

```
> no_solns_8 := (2151-1152) ### the number of ODD primitive solutions
               + 1152 * (1 + 13);
```

$$no\_solns\_8 := 17127 \tag{8}$$

```
>
```

__________[end of s = 8]

• COMPLETE to here