

[> ### Example 7.8. $p = 331$ 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 $n = p^\alpha q_1^{\beta_1} q_2^{\beta_2} \dots q_s^{\beta_s}$, distinct primes $p, q_1, q_2, \dots, q_s = 1, -1, -1, \dots, -1 \pmod{3}$ we seek solutions of the Gauss factorial congruence

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

In this Maple-to-pdf conversion we exhibit all solutions of our paper's congruence (2.5) for the prime $p = 331$, a specially chosen (because of the very large number of support primes) standard Jacobi prime.

It should be emphasised that for $p = 331$ the ' $s = 10$ ' 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 $s = 11$ would require knowing the complete factorisation of the 1291-digit $331^{2^9} + 1$ (that '9' being $11 - 2$).

Note. At $p = 331$ we have $\text{ord}_p \left(\frac{1}{3} \{p-1\} \right)! = (3)$ and $\text{ord}_p \left(\frac{1}{6} \{p-1\} \right)! = 3.2^1$.

No odd square support, so $q = 2$ is the only support which will feature in non-primitive solutions (coming from a base even primitive one).

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.

```

$$\left[\begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \right]$$

(2.1)

(2.2)

(2.3)

```
> 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 10 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]

[[3], 0, 0]

[[4], 0, 0]

[[5], 0, 0]

[[6], 0, 0, 0, 0]

[[7], 0, 0, 0, 0]

[[8], 0, 0, 0, 0, 0, 0]

[[9], 0, 0]

(2.4)

(2.3)

(2.4)

➤

```
p := 331;
```

```
fac := PRFAC(1, (p-1)/3, p, 1);
```

$$p := 331$$
$$fac := -32$$

(1)

$s = 2$ has number of supports = 4. There was **one** (odd) primitive solution, the **5**-digit

$$n = 18205 = 5 * 11 * 331$$

➤

p := 331:

```
fac := -32:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
```

LEV := 2:

$$\text{LEVEL} || \text{LEV}[p] := [] :$$

```
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_{potential331} := [2, 5, 11, 83]

[5, 11]

`Here is a solution:`

18205

`which has`, 5, `digits.`

`There were`, 1, `solutions altogether.`

(3.1)

```
[> no_solns||2 := 4 * 10;
```

no_solns2 := 40

(2)

The procedure name 'ILS' stands for 'identify largest solution'.

```

> ILS := proc(SOLNS, m) local N, Even, Odd, i, Me, Mo;

        N := nops(SOLNS);
        Even := [];
        Odd := [];

        for i to N do if op(i, SOLNS) mod 2 = 0 then

                Even := [op(Even), op(i, SOLNS)] else Odd := [op(Odd), op(i, SOLNS)] fi od;

        Me := 2^m * max(Even): ### Note the use of 'm' here
        Mo := max(Odd):

        if Me > Mo then print(``); lprint(`The largest solution is`, Me, `having`, length(Me), `digits.`)

        else print(``); lprint(`The largest solution is`, Mo, `having`, length(Mo), `digits.`) fi; end;

> SOLNS := [95990, 211178, 274730, 604406, 3983585, 6252590, 8763887, 13755698, 90662555, 199457621,
        259482485, 570861467]: ILS(SOLNS, 3);

        `The largest solution is`, 570861467, having, 9, `digits.`

```

$s = 3$ has number of supports = 6. There were 12 (= 6 ODD + 6 EVEN) primitive solutions, the **least** being the **5**-digit

$n = 95990 = 2 * 5 * 29 * 331$, while the **largest** solution, 570861467, has **9** digits

There were 6 even solutions, each generating an extra 3 solutions

```
>
      p := 331:
      fac := -32:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
      LEV := 3:
      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(``);
```

$S_potential_{331} := [2, 5, 11, 83, 29, 1889]$

[2, 5, 29]

`Here is a solution:`

95990

`which has`, 5, `digits.`

[2, 5, 83]

`Here is a solution:`

274730

`which has`, 6, `digits.`

[2, 5, 1889]

`Here is a solution:`

6252590

`which has`, 7, `digits.`

[2, 11, 29]

`Here is a solution:`

211178

`which has`, 6, `digits.`

[2, 11, 83]

`Here is a solution:`

604406

`which has`, 6, `digits.`

[2, 11, 1889]

`Here is a solution:`

13755698

`which has`, 8, `digits.`

[5, 29, 83]

`Here is a solution:`

3983585

`which has`, 7, `digits.`

[5, 29, 1889]

`Here is a solution:`

90662555

`which has`, 8, `digits.`

[5, 83, 1889]

`Here is a solution:`

259482485

`which has`, 9, `digits.`

```

[11, 29, 83]

`Here is a solution:`

8763887

`which has`, 7, `digits.`

[11, 29, 1889]

`Here is a solution:`

199457621

`which has`, 9, `digits.`

[11, 83, 1889]

`Here is a solution:`

570861467

`which has`, 9, `digits.`

`There were`, 12, `solutions altogether.`

```

(4.1)

```

> sort(SOLN_L);
[95990, 211178, 274730, 604406, 3983585, 6252590, 8763887, 13755698, 90662555, 199457621, 259482485, 570861467]

```

(4.2)

```

> seq(length(n), n = sort(SOLN_L));
5, 6, 6, 6, 7, 7, 7, 8, 8, 9, 9, 9

```

(4.3)

```

> L2 := []: for n in SOLN_L do if n mod 2 = 0 then L2 := [op(L2), n] fi od; L2;
[95990, 274730, 6252590, 211178, 604406, 13755698]

```

(4.4)

```

> [nops(L2)];
seq(length(n), n = sort(L2));

```

```

[6]
5, 6, 6, 6, 7, 8

```

(4.5)

```

> L2sort := sort(L2):

```

```

> MAX := max(SOLN_L); length(MAX);
MAX:=570861467
9

```

(4.6)

```

> MAX_even := 13755698:
2^3*MAX_even;
is(MAX > 2^3*MAX_even);

```

```

110045584
true

```

(4.7)

```

> for n in L2sort do print(``); ifactor(n); seq(PI3(2^j*n), j = 1..4); od;

```

```

(2) (5) (29) (331)
1, 1, 1, -767919

```



```

(2) (11) (29) (331)
1, 1, 1, -1689423

(2) (5) (83) (331)
1, 1, 1, -2197839

(2) (11) (83) (331)
1, 1, 1, -4835247

(2) (5) (331) (1889)
1, 1, 1, -50020719

(2) (11) (331) (1889)
1, 1, 1, -110045583

```

(4.8)

```

> PI3(2^3 * 95990);
PI3(2^4 * 95990);

```

```

1
-767919

```

(4.9)

```

> ILS(SOLN_L, 3);

```

```

`The largest solution is`, 570861467, having, 9, `digits.`

```

```

[ > no_solns||3 := 6 + 6*4;    ### '6' ODD, '6' EVEN
                                no_solns3 := 30

```

(3)

$s = 4$ has number of supports = 8. There were 15 (= 5 ODD + 10 EVEN) primitive solutions, the least being the 8-digit

$n = 13381006 = 2 * 17 * 29 * 41 * 331$, while the largest solution, $331.2^5.41.83.1889$, has 11 digits

There were 10 even solutions, each generating an extra 4 solutions

```

>
p := 331:
fac := -32:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
LEV := 4:
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_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41]

[2, 17, 29, 41]

`Here is a solution:`

13381006

`which has`, 8, `digits.`

[2, 17, 29, 83]

`Here is a solution:`

27088378

`which has`, 8, `digits.`

[2, 17, 29, 1889]

`Here is a solution:`

616505374

`which has`, 9, `digits.`

[2, 17, 41, 83]

`Here is a solution:`

38297362

`which has`, 8, `digits.`

[2, 17, 41, 1889]

`Here is a solution:`

871611046

`which has`, 9, `digits.`

[2, 17, 83, 1889]

`Here is a solution:`

1764480898

`which has`, 10, `digits.`

[2, 29, 41, 83]

`Here is a solution:`

65330794

`which has`, 8, `digits.`

[2, 29, 41, 1889]

`Here is a solution:`

1486865902

`which has`, 10, `digits.`

[2, 29, 83, 1889]

`Here is a solution:`

3009996826

`which has`, 10, `digits.`

[2, 41, 83, 1889]

`Here is a solution:`

4255512754

`which has`, 10, `digits.`

[17, 29, 41, 83]

`Here is a solution:`

555311749

`which has`, 9, `digits.`

[17, 29, 41, 1889]

`Here is a solution:`

12638360167

`which has`, 11, `digits.`

```

[17, 29, 83, 1889]

`Here is a solution:`

25584973021

`which has`, 11, `digits.`

[17, 41, 83, 1889]

`Here is a solution:`

36171858409

`which has`, 11, `digits.`

[29, 41, 83, 1889]

`Here is a solution:`

61704934933

`which has`, 11, `digits.`

`There were`, 15, `solutions altogether.`

```

(5.1)

```

> sort(SOLN_L);
[13381006, 27088378, 38297362, 65330794, 555311749, 616505374, 871611046, 1486865902, 1764480898, 3009996826, 4255512754,
12638360167, 25584973021, 36171858409, 61704934933]

```

(5.2)

```

> seq(length(n), n = sort(SOLN_L));
8, 8, 8, 8, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 11

```

(5.3)

```

> L2 := []: for n in SOLN_L do if n mod 2 = 0 then L2 := [op(L2), n] fi od; L2;
[13381006, 27088378, 616505374, 38297362, 871611046, 1764480898, 65330794, 1486865902, 3009996826, 4255512754]

```

(5.4)

```

> [nops(L2)];
seq(length(n), n = sort(L2));
[10]
8, 8, 8, 8, 9, 9, 10, 10, 10

```

(5.5)

```

> L2sort := sort(L2):

```

```

> MAX := max(SOLN_L); length(MAX);
MAX:=61704934933
11

```

(5.6)

```

> MAX_even := 4255512754:
2^4*MAX_even;
is(MAX > 2^4*MAX_even);
68088204064
false

```

(5.7)

```

> PI3(2^4*MAX_even); PI3(2^5*MAX_even);
1
-68088204063

```

(5.8)

```

> length(2^4*MAX_even);
11

```

(5.9)

```

> ifactor(2^4*MAX_even);
(2)^5 (41) (83) (331) (1889)

```

(5.10)

```
> for n in L2sort do print(``); ifactor(n); seq(Pi3(2^j*n), j = 1..5); od;
```

```
(2) (17) (29) (41) (331)
1, 1, 1, 1, -214096095
```

```
(2) (17) (29) (83) (331)
1, 1, 1, 1, -433414047
```

```
(2) (17) (41) (83) (331)
1, 1, 1, 1, -612757791
```

```
(2) (29) (41) (83) (331)
1, 1, 1, 1, -1045292703
```

```
(2) (17) (29) (331) (1889)
1, 1, 1, 1, -9864085983
```

```
(2) (17) (41) (331) (1889)
1, 1, 1, 1, -13945776735
```

```
(2) (29) (41) (331) (1889)
1, 1, 1, 1, -23789854431
```

```
(2) (17) (83) (331) (1889)
1, 1, 1, 1, -28231694367
```

```
(2) (29) (83) (331) (1889)
1, 1, 1, 1, -48159949215
```

```
(2) (41) (83) (331) (1889)
1, 1, 1, 1, -68088204063
```

(5.11)

```
> ILS(SOLN_L, 4);
```

```
`The largest solution is`, 68088204064, having, 11, `digits.`
```

```
> ifactor(68088204064);
```

```
(2)^5 (41) (83) (331) (1889)
```

(5.12)

```
> no_solns||4 := 5 + 10 * 5; ### '5' ODD solns, '10' EVEN
no_solns4 := 55
```

(4)

$s = 5$ has number of supports = 10. There were 120 (= 60 ODD + 60 EVEN) primitive solutions, the least being the 8-digit

$$n = 17950130 = 2 * 5 * 11 * 17 * 29 * 331$$

the largest, 2109321438652982889909929, has 25 digits

There were 60 even solutions, each generating an extra 5 solutions

```
>
p := 331:

fac := -32:  ### THE ABOVE PRE-COMPUTED VALUE OF fac

LEV := 5:

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 := []:

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_):

SOLN_L := [op(SOLN_L), p*w]:

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(``);
```

$S_{potential331} := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377]$

`There were`, 120, `solutions altogether.`

`The smallest solution is:`

17950130

`The largest solution is:`

2109321438652982889909929

(6.1)

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

8, 8, 8, 8, 8, 9, 9, 9, 10, 10, 10, 10, 10, 10, 11, 11, 11, 11, 11, 12, 14, 14, 14, 14, 14, 14, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15,
15, 15, 15, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17,
17, 17, 17, 17, 17, 17, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 19, 19, 21, 21, 21, 21, 22, 22, 22, 23, 23, 23,
23, 23, 23, 24, 24, 24, 24, 24, 25, 25

(6.2)

```
> L2 := []: for n in SOLN_L do if n mod 2 = 0 then L2 := [op(L2), n] fi od; L2;
[17950130, 25377770, 51374510, 1169234330, 43291490, 87638870, 1994576210, 123903230, 2819918090, 5708614670,
24358654407830, 108372077329910, 34438097611070, 153215695535390, 69716148822410, 310168359254570,
1586672350910030, 7059132899179310, 58747342983590, 261367951207430, 118927547991170, 529110730493090,
2706676363317110, 12042050239776470, 168138947159930, 748053101731610, 3826680375724190, 17024967580373630,
7746694419148970, 3446517827243690, 53589039697226, 238418570125802, 75763814744354, 337074530177858,
```

(6.3)

```
> [nops (L2) ] ;
```

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

[60]

8, 8, 8, 8, 9, 10, 10, 10, 10, 14, 14, 14, 14, 14, 14, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16, 16, 16, 17, 17, 17, 17, 17, 17, 17, 17, 17, 21, 21, 21, 21, 22, 22, 23, 23, 23, 24

(6.4)

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

```
> sort(SOLN_L);
```

[17950130, 25377770, 43291490, 51374510, 87638870, 123903230, 367977665, 744930395, 1053177455, 1169234330, 1796596835, 1994576210, 2819918090, 5708614670, 16953897785, 23969303765, 40888812305, 48523224695, 82774912715, 117026600735, 24358654407830, 34438097611070, 53589039697226, 58747342983590, 69716148822410, 75763814744354, 108372077329910, 118927547991170, 129244154563898, 153215695535390, 153375527409302, 168138947159930, 238418570125802, 261367951207430, 261640605580574, 310168359254570, 337074530177858, 369905683751846, 499352415360515, 529110730493090, 575009492656346, 682370390360054, 748053101731610, 1010884157924945, 1098575313793133, 1164043607084798, 1429181050859405, 1586672350910030, 1645716823809542, 2221627585263155, 2223945147434879, 2438014733818985, 2706676363317110, 3144198311890691, 3490679172002066, 3826680375724190, 4497441209191265, 4887580687578941, 5363632414401767, 5954687999297642, 6358451364718685, 7059132899179310, 7746694419148970, 8418696826593218, 9894370660220783, 10846769975108345, 12042050239776470, 13988593002381107, 15530092378194482, 17024967580373630, 17042727722127734, 23006749088195435, 23862893945238359, 26492510527508234, 32526783193655615, 34465178272463690, 37454928676821986, 50614847994029957, 55486865448000755, 65846902562766245, 71558923026042353, 75823392199420118, 102357427038099995, 112327069077660065, 122071103985601661, 144712224433175855, 144863185638085739, 158807235592553885, 225186339483819989, 246862029915417635, 247119551970852143, 292954015315941365, 318366893752986881, 349375918303618547, 499745084950723505, 543096465813918797, 644498833695071003, 706536154585505645, 1099439186891591711, 1554379540088112419, 323538356218221976382, 457416296722313828678, 780298388526300060686, 925989088486635311714, 1579628445065436708218, 2233267801644238104722, 6632536302473550515831, 13426841783056212019853, 18982776313976023890137, 21074619134352459082262, 32382383123841452518469, 35950820876248312552094, 50827022618144166021926, 102893728714779653166338, 30581977448110656692799, 432029692254225411186371, 736991827963090407317927, 874596694075627051913873, 1491959066364304970911901, 2109321438652982889909929]

(6.5)

```
> MAX := max(SOLN_L); length(MAX);
```

$MAX:=2109321438652982889909929$

25

(6.6)

```
> MAX_even := 50827022618144166021926:
```

$$2^5 * \text{MAX_even};$$

```
is (MAX > 2^5*MAX_even) ;
```

1626464723780613312701632

true

(6.7)

```
> PI3(2^5*MAX even); PI3(2^6*MAX even);
```

1

-1626464723780613312701631

(6.8)

```
> for n in L2sort do print(``); ifactor(n); seq(PI3(2^j*n), j = 1..6); od;
```

(2) (5) (11) (17) (29) (331)

1, 1, 1, 1, 1, -574404159

(2) (5) (11) (17) (41) (331)

1, 1, 1, 1, 1, -812088639

(2) (5) (11) (29) (41) (331)

1, 1, 1, 1, 1, -1385327679

(2) (5) (11) (17) (83) (331)
1, 1, 1, 1, 1, -1643984319

(2) (5) (11) (29) (83) (331)
1, 1, 1, 1, 1, -2804443839

(2) (5) (11) (41) (83) (331)
1, 1, 1, 1, 1, -3964903359

(2) (5) (11) (17) (331) (1889)
1, 1, 1, 1, 1, -37415498559

(2) (5) (11) (29) (331) (1889)
1, 1, 1, 1, 1, -63826438719

(2) (5) (11) (41) (331) (1889)
1, 1, 1, 1, 1, -90237378879

(2) (5) (11) (83) (331) (1889)
1, 1, 1, 1, 1, -182675669439

(2) (5) (17) (29) (331) (14927201)
1, 1, 1, 1, 1, -779476941050559

(2) (5) (17) (41) (331) (14927201)
1, 1, 1, 1, 1, -1102019123554239

(2) (11) (17) (29) (331) (14927201)
1, 1, 1, 1, 1, -1714849270311231

(2) (5) (29) (41) (331) (14927201)
1, 1, 1, 1, 1, -1879914975474879

(2) (5) (17) (83) (331) (14927201)
1, 1, 1, 1, 1, -2230916762317119

(2) (11) (17) (41) (331) (14927201)
1, 1, 1, 1, 1, -2424442071819327

(2) (5) (17) (29) (331) (66411377)
1, 1, 1, 1, 1, -3467906474557119

(2) (5) (29) (83) (331) (14927201)
1, 1, 1, 1, 1, -3805681535717439

(2) (11) (29) (41) (331) (14927201)
1, 1, 1, 1, 1, -4135812946044735

(2) (5) (17) (41) (331) (66411377)
1, 1, 1, 1, 1, -4902902257132479

(2) (11) (17) (83) (331) (14927201)
1, 1, 1, 1, 1, -4908016877097663

(2) (5) (41) (83) (331) (14927201)
1, 1, 1, 1, 1, -5380446309117759

(2) (11) (17) (29) (331) (66411377)
1, 1, 1, 1, 1, -7629394244025663

(2) (5) (29) (41) (331) (66411377)
1, 1, 1, 1, 1, -8363774438637759

(2) (11) (29) (83) (331) (14927201)
1, 1, 1, 1, 1, -8372499378578367

(2) (5) (17) (83) (331) (66411377)
1, 1, 1, 1, 1, -9925387496146239

(2) (11) (17) (41) (331) (66411377)
1, 1, 1, 1, 1, -10786384965691455

(2) (11) (41) (83) (331) (14927201)
1, 1, 1, 1, 1, -11836981880059071

(2) (5) (29) (83) (331) (66411377)
1, 1, 1, 1, 1, -16931543375778879

(2) (11) (29) (41) (331) (66411377)
1, 1, 1, 1, 1, -18400303765003071

(2) (11) (17) (83) (331) (66411377)
1, 1, 1, 1, 1, -21835852491521727

(2) (5) (41) (83) (331) (66411377)
1, 1, 1, 1, 1, -23937699255411519

(2) (11) (29) (83) (331) (66411377)
1, 1, 1, 1, 1, -37249395426713535

(2) (5) (17) (331) (14927201) (1889)
1, 1, 1, 1, 1, -50773515229120959

(2) (11) (41) (83) (331) (66411377)
1, 1, 1, 1, 1, -52662938361905343

(2) (5) (29) (331) (14927201) (1889)
1, 1, 1, 1, 1, -86613643626147519

(2) (11) (17) (331) (14927201) (1889)
1, 1, 1, 1, 1, -111701733504066111

(2) (5) (41) (331) (14927201) (1889)
1, 1, 1, 1, 1, -122453772023174079

(2) (11) (29) (331) (14927201) (1889)
1, 1, 1, 1, 1, -190550015977524543

(2) (5) (17) (331) (66411377) (1889)
1, 1, 1, 1, 1, -225892252773737919

(2) (5) (83) (331) (14927201) (1889)
1, 1, 1, 1, 1, -247894221412767039

(2) (11) (41) (331) (14927201) (1889)
1, 1, 1, 1, 1, -269398298450982975

(2) (5) (29) (331) (66411377) (1889)
1, 1, 1, 1, 1, -385345607672847039

(2) (11) (17) (331) (66411377) (1889)
1, 1, 1, 1, 1, -496962956102223423

```

(2) (5) (41) (331) (66411377) (1889)
1, 1, 1, 1, 1, -544798962571956159

(2) (11) (83) (331) (14927201) (1889)
1, 1, 1, 1, 1, -545367287108087487

(2) (11) (29) (331) (66411377) (1889)
1, 1, 1, 1, 1, -847760336880263487

(2) (5) (83) (331) (66411377) (1889)
1, 1, 1, 1, 1, -1102885704718838079

(2) (11) (41) (331) (66411377) (1889)
1, 1, 1, 1, 1, -1198557717658303551

(2) (11) (83) (331) (66411377) (1889)
1, 1, 1, 1, 1, -2426348550381443775

(2) (17) (29) (331) (66411377) (14927201)
1, 1, 1, 1, 1, -10353227398983103244223

(2) (17) (41) (331) (66411377) (14927201)
1, 1, 1, 1, 1, -14637321495114042517695

(2) (29) (41) (331) (66411377) (14927201)
1, 1, 1, 1, 1, -24969548432841601941951

(2) (17) (83) (331) (66411377) (14927201)
1, 1, 1, 1, 1, -29631650831572329974847

(2) (29) (83) (331) (66411377) (14927201)
1, 1, 1, 1, 1, -50548110242093974662975

(2) (41) (83) (331) (66411377) (14927201)
1, 1, 1, 1, 1, -71464569652615619351103

(2) (17) (331) (66411377) (14927201) (1889)
1, 1, 1, 1, 1, -674387812299278690632383

(2) (29) (331) (66411377) (14927201) (1889)
1, 1, 1, 1, 1, -1150426268039946001667007

(2) (41) (331) (66411377) (14927201) (1889)
1, 1, 1, 1, 1, -1626464723780613312701631

(2) (83) (331) (66411377) (14927201) (1889)
1, 1, 1, 1, 1, -3292599318872948901322815

```

(6.9)

```
> ILS(SOLN_L, 5);
```

```
`The largest solution is`, 3292599318872948901322816, having, 25, `digits.`
```

```
> ifactor(3292599318872948901322816);
```

```
(2)6 (83) (331) (66411377) (14927201) (1889)
```

(6.10)

```
[> no_solns||5 := 60 + 60 * 6; ### '60' ODD solns, '60' EVEN
no_solns5 := 420
```

(5)

For the remaining s-values, for COUNTING (rather than visual examination) purposes, I divide solutions into 2 groups:

1. Solutions which have DON'T have 2 as supports ('false' for '2')

2. Solutions which DO have 2 as support ('true' for '2')

_____ [start of s = 6:]

```
[> s := 6: Pow(331^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
7
```

(6)

(i, false) s = 6 has number of supports = 14. There were 561 (odd) primitive solutions, the least having 13 digits,

the largest 48 digits

```
>
p := 331:
fac := -32: ### 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];
print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
count := 0:
SOLN_L := []:
for L_ in choose(S_potential||p, LEV) do
if member(2, L_) = false
and PI3_more_modified(L_, LEV, p, fac) = 1 then
count := count+1:
SOLN_L := [op(SOLN_L), p*multiplicity(L_)]:
fi; od:
print(``); lprint(`and there were`, count, `primitive solutions.`);
MIN_soln := min(sort(SOLN_L)):
```

```

print(``); MIN_soln := min(sort(SOLN_L));
print(``); MAX_soln := max(sort(SOLN_L));

print(_____); print(``);

S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849]

`Here there are`, 14, `support primes.`
`and there were`, 561, `primitive solutions.`

MIN_soln := 5244919469305

MAX_soln := 747370295182576589170858273400643916856455312201

```

(7.1)

```

> length(MIN_soln); length(MAX_soln);

```

13
48

(7.2)

```

> no_solns6_11 := 561;

```

no_solns6_1 := 561

(7)

(ii, true) $s = 6$ has number of supports = 14. There were 420 (even) primitive solutions, and thus $7 \cdot 420$ EVEN solutions

altogether; the **least** solution has **10** digits, and the **largest** (having a highest-factor-of-2: 2^7) has **47** digits

```

>
    p := 331:
    fac := -32: ### 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];
    print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
    count := 0:
    SOLN_L := []:
    for L_ in choose(S_potential||p, LEV) do
        if member(2, L_) = true
        and PI3_more_modified(L_, LEV, p, fac) = 1 then
            count := count+1:
            SOLN_L := [op(SOLN_L), p*mul(j, j = L_)]:
        fi; od:
    print(``); lprint(`and there were`, count, `EVEN primitive solutions.`);

```

```

MIN_soln := min(sort(SOLN_L));
print(``); MIN_soln := min(sort(SOLN_L));
print(``); MAX_soln := max(sort(SOLN_L));

print(_____); print(``);

```

S_{potential331} := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849]

`Here there are`, 14, `support primes.`

`and there were`, 420, `EVEN primitive solutions.`

MIN_soln := 5553117490

MAX_soln := 33529868889069552337011202619797209495407606

```
[> length(MIN_soln);
```

10

```
[> MAX := 2^6 * MAX_soln;
```

MAX := 21459116088900451349568716967667021407706086784

```
[> length(MAX);
```

47

```
[> ifactor(MAX);
```

(2)⁷ (331) (1776833) (6271510529) (18581275406849) (66411377) (36833)

```
[> PI3(MAX);
```

1

```
[> ILS(SOLN_L, 6);
```

*`The largest solution is`, 21459116088900451349568716967667021407706086784, having, 47,`
`digits.`*

```
[> ifactor(21459116088900451349568716967667021407706086784);
```

(2)⁷ (331) (1776833) (6271510529) (18581275406849) (66411377) (36833)

```
[> no_solns6_||2 := 7*420;
```

no_solns6_2 := 2940

```
[> TOTAL||6 := add(no_solns6_||i, i = 1..2);
```

TOTAL6 := **3501**

_____ [end of s = 6, start of s = 7:]

```
[> s := 7: Pow(331^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
```

8

(i, false) $s = 7$ has number of supports = 18. There were 6517 (odd) primitive solutions, the least having 15 digits,

the largest 117 digits

```
>
      p := 331:
      fac := -32:  ### 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];
      print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
      count := 0:
      SOLN_L := []:
      for L_ in choose(S_potential||p, LEV) do
          if member(2, L_) = false
          and PI3_more_modified(L_, LEV, p, fac) = 1 then
              count := count+1:
              SOLN_L := [op(SOLN_L), p*multiplication(j, j = L_)]:
          fi; od:
      print(``); lprint(`and there were`, count, `primitive solutions.`);
      MIN_soln := min(sort(SOLN_L)):
      print(``); MIN_soln := min(sort(SOLN_L));
      print(``); MAX_soln := max(sort(SOLN_L));
      print(_____); print(``);

S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977,
26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993]

      `Here there are`, 18, `support primes.`
      `and there were`, 6517, `primitive solutions.`

      MIN_soln := 946104062682515

MAX_soln :=
26602532671381122654949060782629957559652790425278685176807986462090387035160801615514662929561115432929269\
0176023931

_____

> length(MIN_soln); length(MAX_soln);

      15
      117
```

(9.1)

(9.2)

```
[> no_solns7_||1 := 6517;
```

```
no_solns7_1 := 6517
```

(11)

(ii, true) $s = 7$ has number of supports = 18. There were **4021 (even) primitive solutions**, and thus **8*4021 EVEN solutions**

altogether; the **least** solution has **14** digits, and the **largest** (having a highest-factor-of-2: 2^8) has **109** digits

```
>
    p := 331:
    fac := -32: ### 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];
    print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
        count := 0:
        SOLN_L := []:
    for L_ in choose(S_potential||p, LEV) do
        if member(2, L_) = true
        and PI3_more_modified(L_, LEV, p, fac) = 1 then
            count := count+1:
            SOLN_L := [op(SOLN_L), p*mul(j, j = L_)]:
    fi; od:
    print(``); lprint(`and there were`, count, `EVEN primitive solutions.`);
        MIN_soln := min(sort(SOLN_L)):
    print(``); MIN_soln := min(sort(SOLN_L));
    print(``); MAX_soln := max(sort(SOLN_L));

    print(_____); print(``);
```

```
S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977,
26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993]
```

```
`Here there are`, 18, `support primes.`
```

```
`and there were`, 4021, `EVEN primitive solutions.`
```

```
MIN_soln := 22797688257410
```

```
MAX_soln :=
```

```
19922982266270772723631998676998140855409808943198769461462672641801789674517807617553711935259685384461814
```

```
[> length(MIN_soln);
```

(10.1)

14 (10.2)

```
> MAX := 2^7 * MAX_soln;
MAX:=
```

(10.3)

```
25501417300826589086248958306557620294924555447294424910672220981506290783382793750468751277132397292111121\
92
```

```
> length(MAX);
```

109 (10.4)

```
> PI3(2^7 * MIN_soln);
```

1 (10.5)

```
> ifactor(2^7 * MIN_soln);
```

(2)⁸ (5) (11) (17) (29) (41) (331) (30977) (10.6)

```
> PI3(2^8 * MIN_soln);
```

-2918104096948479 (10.7)

```
> ILS(SOLN_L, 7);

`The largest solution is`,
2550141730082658908624895830655762029492455544729442491067222098150629078338279375046875127713
239729211112192, having, 109, `digits.`
```

```
> no_solns7_||2 := 8*4021;
```

no_solns7_2:=32168 (12)

```
> TOTAL||7 := add(no_solns7_||i, i = 1..2);
```

TOTAL7:=38685 (13)

_____ [end of s = 7, start of s = 8:]

```
> s := 8: Pow(331^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
```

9 (14)

(i, false) s = 8 has number of supports = 24. There were 163 821 (odd) primitive solutions, the least having 17 digits, the largest 130 digits, and the time taken was 50- secs

```
>
st := time[real]():
p := 331:
fac := -32: ### 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:
S_potential||p := LEVEL||LEV[p];
print(''); lprint('Here there are', nops(S_potential||p), 'support primes.');
```

```

count := 0:

MIN := 10^100: ### Clearly greater than ANY small solution

MAX := 0:

for L_ in choose(S_potential||p, LEV) do
    if member(2, L_) = false
and PI3_more_modified(L_, LEV, p, fac) = 1 then
count := count+1:

w := mul(q, q = L_):

MIN := min(MIN, p*w):

MAX := max(MAX, p*w):

fi; od:

print(''); lprint('and there were', count, 'primitive solutions.');
```

```

print(''); lprint('The smallest solution is:'); print(''); MIN;
print(''); lprint('The largest solution is:'); print(''); MAX;
print(''); lprint('The time taken to perform that computation was:');
```

```

print(''); time[real]() - st;
print(''); lprint('seconds.');
```

```

S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977,
26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993, 641, 3329, 4481, 51713,
1024640129, 20973135548033]

`Here there are`, 24, `support primes.`

`and there were`, 163821, `primitive solutions.`

`The smallest solution is:`

36981927178069555

`The largest solution is:`

5579385236378527093888158618770023828299120580052749719724464426492053142694583170306867118565280340015757786120\
184543330607977723

`The time taken to perform that computation was:`

48.953

`seconds.`
```

```

> length(MIN);
length(MAX);

17
130

> PI3(MIN);
1
```

(11.1)

(11.2)

(11.3)

```
> no_solns8_||1 := 163821;
```

no_solns8_1 := **163821**

(15)

(ii, true) $s = 8$ has number of supports = **24**. There were **81936 (even)** primitive solutions, and thus **9*81936 EVEN solutions**

altogether; the **least solution has 14 digits**, and the **largest (having a highest-factor-of-2: 2^9) has 123 digits**, and the time taken was **32.7 secs**

```
>
    st := time[real]():
    p := 331:
    fac := -32: ### 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:
    S_potential||p := LEVEL||LEV[p]:
    print(''); lprint('Here there are', nops(S_potential||p), 'support primes.'):
    count := 0:
    MIN := 10^100: ### Clearly greater than ANY small solution
    MAX := 0:
    for L_ in choose(S_potential||p, LEV) do
        if member(2, L_) = true
and PI3_more_modified(L_, LEV, p, fac) = 1 then
            count := count+1:
            w := mul(q, q = L_):
            MIN := min(MIN, p*w):
            MAX := max(MAX, p*w):
        fi; od:
    print(''); lprint('and there were', count, 'EVEN primitive solutions.'):
    print(''); lprint('The smallest solution is:'); print(''); MIN;
    print(''); lprint('The largest solution is:'); print(''); MAX;
    print(''); lprint('The time taken to perform that computation was:');
    print(''); time[real]() - st;
    print(''); lprint('seconds.'): print('');
```

S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977, 26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993, 641, 3329, 4481, 51713, 1024640129, 20973135548033]

'Here there are', 24, 'support primes.'

`and there were`, 81936, `EVEN primitive solutions.`

`The smallest solution is:`

39155031421990

`The largest solution is:`

3267693355056378926815614112847694659442208419550055082040579360735294798659741852940522931729539887858888993650\979030934

`The time taken to perform that computation was:`

32.698

`seconds.`

(12.1)

> length(MIN);

length(MAX);

14

121

(12.2)

> PI3(2^8 * MIN); PI3(2^9 * MIN);

1

-10023688044029439

(12.3)

> ifactor(2^8 * MIN);

(2)^9 (5) (11) (17) (29) (41) (83) (331) (641)

(12.4)

> TRUE_MAX := 2^8 * MAX;

length(TRUE_MAX);

123

(12.5)

> no_solns8_||2 := 9*81936;

no_solns8_2:=737424

(16)

> TOTAL||8 := add(no_solns8_||i, i = 1..2);

TOTAL8 := 901245

(17)

_____ [end of s = 8, start of s = 9:]

> s := 9: Pow(331^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution

10

(18)

(i, false) s = 9 has number of supports = 26. There were 681 180 (odd) primitive solutions, the least having 19 digits,

the largest 142 digits, and the time taken was 264 secs

```

> st := time[real]():

p := 331:

fac := -32: ### THE ABOVE PRE-COMPUTED VALUE OF fac

LEV := 9:

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:
S_potential||p := LEVEL||LEV[p];
print(''); lprint('Here there are', nops(S_potential||p), 'support primes.');
```

```

count := 0:

MIN := 10^100: ### Clearly greater than ANY small solution

MAX := 0:

for L_ in choose(S_potential||p, LEV) do
    if member(2, L_) = false
and PI3_more_modified(L_, LEV, p, fac) = 1 then
count := count+1:

w := mul(q, q = L_):

MIN := min(MIN, p*w):

MAX := max(MAX, p*w):

fi; od:

print(''); lprint('and there were', count, 'primitive solutions.');
```

```

print(''); lprint('The smallest solution is:'); print(''); MIN;
print(''); lprint('The largest solution is:'); print(''); MAX;
print(''); lprint('The time taken to perform that computation was:');
```

```

print(''); time[real]() - st;
print(''); lprint('seconds.');
```

```

S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977,
26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993, 641, 3329, 4481, 51713,
1024640129, 20973135548033, 257, 345807320321]

    'Here there are', 26, 'support primes.'
```

```

    'and there were', 681180, 'primitive solutions.'
```

```

    'The smallest solution is:'

9504355284763875635

    'The largest solution is:'

1929392257630607620762359708829862553025436695028904545405600841792106979476740443445882493535686184488992946932\
412464179160657156135493209083

    'The time taken to perform that computation was:'

264.291

'seconds.'
```

```

> length(MIN);

length(MAX);
```

(13.1)


```

print(``); lprint(`The largest solution is:`); print(``); MAX;
print(``); lprint(`The time taken to perform that computation was:`);
print(``); time[real]() - st;
print(``); lprint(`seconds.`); print(``);
S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977,
26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993, 641, 3329, 4481, 51713,
1024640129, 20973135548033, 257, 345807320321]

`Here there are`, 26, `support primes.`
`and there were`, 360678, `EVEN primitive solutions.`
`The smallest solution is:`
10062843075451430
`The largest solution is:`
1129992282742784412559681286025850988584132774126989056448400561318211498530494552426973489213843014538765245074\
645441025447905809814

`The time taken to perform that computation was:`
178.095
`seconds.`

```

(14.1)

```

> length(MIN);
length(MAX);

```

17
133

(14.2)

```

> PI3(2^9 * MIN); PI3(2^10 * MIN);

```

1
-5152175654631132159

(14.3)

```

> ifactor(2^9 * MIN);

```

(2)¹⁰ (5) (11) (17) (29) (41) (83) (257) (331) (641)

(14.4)

```

> TRUE_MAX := 2^9 * MAX;
length(TRUE_MAX);

```

135

(14.5)

```

> no_solns9_||2 := 10*360678;

```

no_solns9_2 := **3606780**

(20)

```

> TOTAL||9 := add(no_solns9_||i, i = 1..2);

```

TOTAL9 := 4287960

(21)

_____ [end of s = 9, start of s = 10:]

NOTE. There is no extra support prime in passing from level 9 to 10, so there are still 26 supports.

```
[> s := 10: Pow(331^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
11
```

(22)

(i, false) $s = 10$ has number of supports = 26. There were 1 089 324 (odd) primitive solutions, the least having 23 digits,

the largest 150 digits, and the time taken was 112 secs

```
>
    st := time[real]():
    p := 331:
    fac := -32: ### THE ABOVE PRE-COMPUTED VALUE OF fac
    LEV := 10:
    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:
    S_potential||p := LEVEL||LEV[p];
    print(''); lprint('Here there are', nops(S_potential||p), 'support primes.');
```

count := 0:

MIN := 10^100: ### Clearly greater than ANY small solution

MAX := 0:

for L_ in choose(S_potential||p, LEV) do

if member(2, L_) = false

and PI3_more_modified(L_, LEV, p, fac) = 1 then

count := count+1:

w := mul(q, q = L_):

MIN := min(MIN, p*w):

MAX := max(MAX, p*w):

fi; od:

print(''); lprint('and there were', count, 'primitive solutions.');

print(''); lprint('The smallest solution is:'); print(''); MIN;

print(''); lprint('The largest solution is:'); print(''); MAX;

print(''); lprint('The time taken to perform that computation was:');

print(''); time[real]() - st;

print(''); lprint('seconds.');

print('');

S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977, 26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993, 641, 3329, 4481, 51713, 1024640129, 20973135548033, 257, 345807320321]

'Here there are', 26, 'support primes.'

'and there were', 1089324, 'primitive solutions.'

'The smallest solution is:'

```

42589016031026926720435
`The largest solution is:`
2353206951895369492822876605906442958120090998535870209051019728810331528758206339310322534631190464848338298252\
83146695818320219335924952691865240731

```

```

`The time taken to perform that computation was:`

```

```

112.174

```

```

`seconds.`

```

(15.1)

```

> length(MIN) ;
length(MAX) ;

```

```

23
150

```

(15.2)

```

> PI3(MIN) ;

```

```

1

```

(15.3)

```

[> no_solns10_||1 := 1089324;

```

```

no_solns10_1:=1089324

```

(23)

(ii, true) $s = 10$ has number of supports = 26. There were 680 822 (even) primitive solutions, and thus $11^* 680822$ EVEN

solutions altogether; the least solution has 20 digits, and the largest (having a highest-factor-of-2: 2^{11}) has 147 digits, and

the time taken was 68 secs

```

> st := time[real]():
p := 331:
fac := -32: ### THE ABOVE PRE-COMPUTED VALUE OF fac
LEV := 10:
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:
S_potential||p := LEVEL||LEV[p];
print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
count := 0:
MIN := 10^100: ### Clearly greater than ANY small solution
MAX := 0:
for L_ in choose(S_potential||p, LEV) do
if member(2, L_) = true

```



```

and PI3_more_modified(L_, LEV, p, fac) = 1 then
    count := count+1:
    w := mul(q, q = L_):
    MIN := min(MIN, p*w):
    MAX := max(MAX, p*w):
fi; od:
print(``); lprint(`and there were`, count, `EVEN primitive solutions.`);
print(``); lprint(`The smallest solution is:`); print(``); MIN;
print(``); lprint(`The largest solution is:`); print(``); MAX;
print(``); lprint(`The time taken to perform that computation was:`);
print(``); time[real]() - st;
print(``); lprint(`seconds.`); print(``);
S_potential331 := [2, 5, 11, 83, 29, 1889, 17, 41, 14927201, 66411377, 36833, 1776833, 6271510529, 18581275406849, 30977,
26705372033, 226515295026304671802528341454337, 1150085914749541327603538276348993, 641, 3329, 4481, 51713,
1024640129, 20973135548033, 257, 345807320321]

`Here there are`, 26, `support primes.`
`and there were`, 680822, `EVEN primitive solutions.`
`The smallest solution is:`
45091599821097857830
`The largest solution is:`
2648765474183977326876388052120196955054404098230807519953782633269791284558769832967220368634745600569483230278\
62659795060447683418672382482614
`The time taken to perform that computation was:`
67.954
`seconds.`

```

(16.1)

```

> length(MIN);
length(MAX);

```

20
144

(16.2)

```

> PI3(2^10 * MIN); PI3(2^11 * MIN);

```

1
-46173798216804206417919

(16.3)

```

> ifactor(2^10 * MIN);

```

(2)¹¹ (5) (11) (17) (29) (41) (83) (257) (331) (641) (4481)

(16.4)

```

> TRUE_MAX := 2^10 * MAX:

```

```
length(TRUE_MAX);
```

147

(16.5)

```

> no_solns10_||2 := 11*680822;

```

no_solns10_2 := 7489042

(24)

```

> TOTAL||10 := add(no_solns10_||i, i = 1..2);

```

(25)

L

TOTAL10 := 8578366

(25)

_____ [end of s = 10]

• COMPLETE TO HERE (Wed/Thurs 20th/21st May 2015)