```
> ### EXTENDED Example 1. p = 7 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 $p = 7$, the very first Jacobi prime.

"EXTENDED" means this: in our paper - purely because of space constraints - we mentioned only solutions up to $s = 6$, while here in this Maple worksheet we extend to $s = 11$. It should be emphasised that for $p = 7$ the '$s = 11$' 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 $s = 12$ would require knowing the complete factorisation of the 866-digit $7^{2^{10}} + 1$ (that '10' being $12 - 2$)

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 $p = 7$

```
>              p := 7:
     level||1[p] := [2]:                             ### These divide (7 - 1)
     level||2[p] := []:                              ### These divide (7 + 1)
     level||3[p] := [5]:                             ### These divide (7^2 + 1)
     level||4[p] := []:                              ### No 'q' for 7^(2^2) + 1
     level||5[p] := [17, 169553]:                    ### These divide 7^(2^3) + 1
     level||6[p] := [353, 47072139617]:              ### These divide 7^(2^4)
     level||7[p] := [7699649, 531968664833]:         ### These divide 7^(2^5) + 1
     level||8[p] := [35969, 1110623386241]:          ### These divide 7^(2^6) + 1
     level||9[p] := [257, 197231873]:                ### These divide 7^(2^7) + 1
     level||10[p] := [28667393, 126575155274810369]: ### These divide 7^(2^8) + 1
                                                     ### The 2nd of those level 9s
                                                     ### at Wilfrid's site
     level||11[p] := [13313, 275102002206713516320479233, 133833088877063359811677099009,
                  656929861401793262700329631944023570433]:
                                                     ### These divide 7^(2^9) + 1
                                                     ### The 2nd, 3rd and 4th of those
                                                     ### level 10s at Wilfrid's site
                           ### COMPLETE TO HERE
              LEV := 11:
    LEVEL||LEV[p] := []:
   for lev from 1 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, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369, 13313, 275102002206713516320479233, 133833088877063359811677099009, 656929861401793262700329631944023570433]

(2.1)

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

true, true, true, true, true, true, true, true, true, true, true, true, true, true, true, true, true, true **(2.2)**

```
> nops(LEVEL||LEV[p]);
```

18 **(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, 2, 2, 2 **(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 11 do if level||LEV[p] <> [] then
  print([[LEV], seq(p&^(2^(LEV - 2)) + 1 mod q, q = level||LEV[p])]); fi od;
```

[[1], 0]
[[3], 0]
[[5], 0, 0]
[[6], 0, 0]
[[7], 0, 0]
[[8], 0, 0]
[[9], 0, 0]
[[10], 0, 0]
[[11], 0, 0, 0, 0] **(2.5)**

```
>            p := 7;
          fac := PRFAC(1, (p-1)/3, p, 1);
```

$p := 7$
$fac := 2$ **(1)**

• **No solutions at $s = 2, 3, 4$ or 5 (not enough supports)**

• **The singleton at $s = 6$ (where there are 6 supports) did not generate a solution**

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

9 **(2)**

$s = 7$ has number of supports = 8. There were 3 (all even) primitive solutions, the least being the 30-digit

$n = 291732036793260341339438944070 = 2 * 5 * 17 * 353 * 169553 * 7699649 * 531968664833 * 7$

**The three even solutions each generate (by theory) 8 further solutions, and, since each of the three solutions also had
a 5-factor, then a further doubling of the number of solutions is obtained**

The least of the 54 solutions is the 30-digit $7 . 2 . 5 . 17 . 353 . 169553 . 7699649 . 531968664833$, **while**

the largest is the 37-digit $7 . 2^9 . 5^2 . 17 . 353 . 7699649 . 47072139617 . 531968664833$

```
>            p := 7:
          fac := 2:   ### 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:
         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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833]$

[2, 5, 17, 353, 169553, 7699649, 531968664833]

`Here is a solution:`

291732036793260341339438944070

`which has`, 30, `digits.`

[2, 5, 17, 353, 169553, 47072139617, 531968664833]

`Here is a solution:`

1783516517010597723631187183170310

`which has`, 34, `digits.`

[2, 5, 17, 353, 7699649, 47072139617, 531968664833]

`Here is a solution:`

80992086053824655135321384839608230

`which has`, 35, `digits.`

`There were`, 3, `solutions altogether.`

(3.1)

```
> n := 291732036793260341339438944070; seq(PI3(2^j*n), j = 1..9);
```

$n := 291732036793260341339438944070$

1, 1, 1, 1, 1, 1, 1, 1, -74683401419074647382896369681919 **(3.2)**

```
> n := 17835165170105977236311871831703l0; seq(PI3(2^j*n), j = 1..9);
```
$$n := 17835165170105977236311871831703l0$$
$$1, 1, 1, 1, 1, 1, 1, 1, -456580228354713017249583918891599359 \quad (3.3)$$

```
> n := 809920860538246551353213848396082 30; seq(PI3(2^j*n), j = 1..9);
```
$$n := 80992086053824655135321384839608230$$
$$1, 1, 1, 1, 1, 1, 1, 1, -2073397402977911171464227451893970 6879 \quad (3.4)$$

```
> n := 17835165170105977236311871831703l0; ifactor(n);

  seq(PI3(2^j*n), j = 1..9);
```
$$n := 17835165170105977236311871831703l0$$
$$(2)\ (5)\ (7)\ (17)\ (353)\ (47072139617)\ (169553)\ (531968664833)$$
$$1, 1, 1, 1, 1, 1, 1, 1, -456580228354713017249583918891599359 \quad (3.5)$$

```
> ifactor(456580228354713017249583918891599360);
```
$$(2)^9\ (5)\ (7)\ (17)\ (353)\ (47072139617)\ (169553)\ (531968664833) \quad (3.6)$$

```
> length(456580228354713017249583918891599360);
```
$$36 \quad (3.7)$$

```
> MAX := 80992086053824655135321384839608230:

  length(MAX * 2*8 * 5);
```
$$37 \quad (3.8)$$

```
> PI3(MAX * 2^8 * 5); ### in an instant
```
$$1 \quad (3.9)$$

```
> PI3(MAX * 2^9 * 5); ### in an instant
```
$$-1036698701488955585732113725946985343 99 \quad (3.10)$$

```
> ifactor(MAX * 2^8 * 5);
```
$$(2)^9\ (5)^2\ (7)\ (17)\ (353)\ (7699649)\ (47072139617)\ (531968664833) \quad (3.11)$$

____________________[end of s = 7]

Now, for the remaining s-values, for counting purposes, we need to divide solutions into 4 groups:

1. Solutions which have NEITHER 2 NOR 5 as supports ('false' for BOTH '2' and '5')

2. Solutions which have 2, but not 5, as support ('true' for '2' and 'false' for '5')

3. Solutions which have 5, but not 2, as support ('false' for '2' and 'true' for '5')

4. Solutions which have BOTH 2 AND 5 as supports ('true' for '2' AND for '5')

____________________[start of s = 8:]

```
> s := 8: Pow(7^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
```
$$10 \quad (3)$$

(i, false & false) s = 8 has number of supports = 10. There were NO primitive solutions (from a singleton)

```
>              p := 7:
            fac := 2:  ### 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];
print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
         count := 0:
        SOLN_L := []:
 print(``); print(____________________); print(``);
for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = false
              and member(5, L_) = false
      and 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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241]$

`Here there are`, 10, `support primes.`

`There were`, 0, `solutions altogether.`

(4.1)

```
> no_solns||1 := 0;
```

$no\_solns1 := 0$ **(4)**

(ii, true & false) **s = 8** **has number of supports = 10. There were 4 primitive solutions, the least having 45 digits,**

the largest (having a highest-factor-of-2: 2^{10}) having 58 digits

```
>              p := 7:
            fac := 2:  ### 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];
print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
         count := 0:
        SOLN_L := []:
 print(``); print(____________________); print(``);
for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = true
              and member(5, L_) = false
      and 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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241]$

`Here there are`, 10, `support primes.`

[2, 17, 353, 35969, 169553, 7699649, 47072139617, 1110623386241]

`Here is a solution:`

206246784151868956757983826214736573915849294

`which has`, 45, `digits.`

[2, 17, 353, 169553, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

3050316283226380306044796028370600227274240691329358

`which has`, 52, `digits.`

[2, 17, 35969, 169553, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

310812539352322020476275547718023001628405562114520334

`which has`, 54, `digits.`

[2, 353, 35969, 169553, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

$$6453930964198216072242662843791889386754539025083863406$$

`which has`, 55, `digits.`

____

`There were`, 4, `solutions altogether.`

____

(5.1)

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

[45, 52, 54, 55] (5.2)

```
> MAX1 := max(SOLN_L);
```

$MAX1 := 6453930964198216072242662843791889386754539025083863406$ (5.3)

```
> seq(PI3(2^j*MAX1), j = 1..10);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, -3304412653669486628988243376021447366018323980842938063871 (5.4)

```
> length(2^9 * MAX1);
```

58 (5.5)

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

$no\_solns2 := 40$ (5)

(iii, false & true) ***s*** **= 8 has number of supports = 10. There were 4 primitive solutions, the least having 45 digits,**

the largest (having a highest-factor-of-5: 5^2) having 56 digits

```
>              p  := 7:
             fac  := 2:   ### 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];
 print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
          count  := 0:
         SOLN_L  := []:
  print(``); print(____________________); print(``);
 for L_  in choose(S_potential||p, LEV) do
               if member(2, L_) = false
              and member(5, L_) = true
       and 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_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241]

`Here there are`, 10, `support primes.`

____

[5, 17, 353, 35969, 169553, 7699649, 47072139617, 1110623386241]

`Here is a solution:`

515616960379672391894959565536841434789623235

`which has`, 45, `digits.`

____

[5, 17, 353, 169553, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

7625790708065950765111990070926500568185601728323395

`which has`, 52, `digits.`

____

[5, 17, 35969, 169553, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

777031348380805051190688869295057504071013905286300835

`which has`, 54, `digits.`

____

[5, 353, 35969, 169553, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

16134827410495540180606657109479723466886347562709658515

`which has`, 56, `digits.`

____

`There were`, 4, `solutions altogether.`

____

(6.1)

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

[45, 52, 54, 56] **(6.2)**

```
> MAX1 := max(SOLN_L);
```

MAX1 := 16134827410495540180606657109479723466886347562709658515 **(6.3)**

```
> seq(PI3(5^j*MAX1), j = 1..2);
```

1, 161348274104955401806066571094797234668863475627096585151 **(6.4)**

```
> length(5 * MAX1);
```

56 **(6.5)**

```
> no_solns||3 := 4 * 2;
```

$$no\_solns3 := 8 \quad (6)$$

(iv, true & true) $s = 8$ has number of supports = 10. There were 7 primitive solutions, the least having 33 digits,

the largest (having a highest-factor-of-2 and 5: $2^{10}.5^{2}$) having 49 digits

```
>               p := 7:
              fac := 2:  ### 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];
            count := 0:
           SOLN_L := []:
    print(``); print(____________________); print(``);
   for L_ in choose(S_potential||p, LEV) do
                 if member(2, L_) = true
                and member(5, L_) = true
         and 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(``);
   SOLN_L;
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241]$

____

$[2, 5, 17, 353, 35969, 169553, 7699649, 47072139617]$

`Here is a solution:`

928518103917876550769579132865670

`which has`, 33, `digits.`

____

[2, 5, 17, 353, 35969, 169553, 531968664833, 1110623386241]

`Here is a solution:`

1513590434540510243301489852440246582470

`which has`, 40, `digits.`

[2, 5, 17, 353, 35969, 7699649, 531968664833, 1110623386241]

`Here is a solution:`

68734348998362784228684087222589350577510

`which has`, 41, `digits.`

[2, 5, 17, 353, 35969, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

420210437194544463472226202769493994679745830

`which has`, 45, `digits.`

[2, 5, 17, 353, 169553, 7699649, 47072139617, 531968664833]

`Here is a solution:`

13732451166684131752159146763710094221190

`which has`, 41, `digits.`

[2, 5, 17, 35969, 169553, 7699649, 47072139617, 531968664833]

`Here is a solution:`

1399270640267596416411933002673904756492870

`which has`, 43, `digits.`

[2, 5, 353, 35969, 169553, 7699649, 47072139617, 531968664833]

`Here is a solution:`

29055443294968325587847785290816963473057830

`which has`, 44, `digits.`

`There were`, 7, `solutions altogether.`

[928518103917876550769579132865670, 1513590434540510243301489852440246582470, 68734348998362784228684087222589350577510, 420210437194544463472226202769493994679745830, 13732451166684131752159146763710094221190, 1399270640267596416411933002673904756492870, 29055443294968325587847785290816963473057830] **(7.1)**

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

[33, 40, 41, 41, 43, 44, 45] **(7.2)**

```
> MAX1 := max(SOLN_L);
```

$$MAX1 := 420210437194544463472226202769493994679745830 \quad (7.3)$$

```
> seq(PI3(2^j*MAX1), j = 1..10);
```

$$1, 1, 1, 1, 1, 1, 1, 1, 1, -2151477438436067652977798158179809252760298649599 \quad (7.4)$$

```
> seq(PI3(2^j* 5 * MAX1), j = 1..10);
```

$$1, 1, 1, 1, 1, 1, 1, 1, 1, -10757387192180338264888990790899046263801493247 99 \quad (7.5)$$

```
> length(2^9 * 5 * MAX1);
```

$$49 \quad (7.6)$$

```
> ifactor(2^9 * 5 * MAX1);
```

$$(2)^{10} (5)^{2} (7) (17) (353) (47072139617) (1110623386241) (531968664833) (35969) \quad (7.7)$$

```
> no_solns||4 := 7 * 10 * 2;
```

$$no\_solns4 := 140 \quad (7)$$

```
> TOTAL||8 := add(no_solns||i, i = 1..4);
```

$$TOTAL8 := 188 \quad (8)$$

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

```
> s := 9: Pow(7^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
```

$$11 \quad (9)$$

(i, false & false) s = 9 has number of supports = 12. There were 4 primitive solutions, having between 55 and 66 digits

```
>              p := 7:
            fac := 2:  ### 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:
print(``); S_potential||p := LEVEL||LEV[p];
print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
         count := 0:
        SOLN_L := []:
 print(``); print(____________________); print(``);
for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = false
              and member(5, L_) = false
      and 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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873]$

`Here there are`, 12, `support primes.`

____

[17, 257, 353, 35969, 169553, 7699649, 197231873, 47072139617, 1110623386241]

`Here is a solution:`

5227179480697228256763410770623010815740828232413174567

`which has`, 55, `digits.`

____

[17, 257, 353, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

77308117801133164994141392021336558793298856642175981762238119

`which has`, 62, `digits.`

____

[17, 257, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

7877324898552291251201902916757661992170443554001212147325617287

`which has`, 64, `digits.`

____

[257, 353, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

163570334658174047745545395859732628425656857327201640470937817783

`which has`, 66, `digits.`

____

`There were`, 4, `solutions altogether.`

____

(8.1)

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

[55, 62, 64, 66] (8.2)

```
> no_solns||1 := 4;
```

$no\_solns1 := 4$ (10)

(ii, true & false) s = 9 has number of supports = 12. There were 14 primitive solutions, the least having 43

digits,

the largest (having a highest-factor-of-2: 2^{11}) having 67 digits

```
>            p := 7:
           fac := 2:  ### 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:
 print(``); S_potential||p := LEVEL||LEV[p];
 print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
         count := 0:
        SOLN_L := []:
 print(``); print(____________________); print(``);
 for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = true
              and member(5, L_) = false
      and 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_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873]

`` `Here there are`, 12, `support primes.` ``

[2, 17, 257, 353, 35969, 169553, 7699649, 197231873, 47072139617]

`` `Here is a solution:` ``

9413054948156755516960673347156134307095374

`` `which has`, 43, `digits.` ``

[2, 17, 257, 353, 35969, 169553, 7699649, 531968664833, 1110623386241]

`` `Here is a solution:` ``

5990215148919774249323601542997769769452836257 42 `which has`, 48, `digits.`

[2, 17, 257, 353, 35969, 169553, 197231873, 531968664833, 1110623386241]

`Here is a solution:`

15344353404868468704628781330566369537779723208334

`which has`, 50, `digits.`

[2, 17, 257, 353, 35969, 169553, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

3662144129243035107136467385010247185388656158299086

`which has`, 52, `digits.`

[2, 17, 257, 353, 35969, 7699649, 197231873, 531968664833, 1110623386241]

`Here is a solution:`

696809465768474165559596654397822607946754749378222

`which has`, 51, `digits.`

[2, 17, 257, 353, 35969, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

166303305648275206098554398710295687665394189430982638

`which has`, 54, `digits.`

[2, 17, 257, 353, 35969, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

4259975027316283913497733033027921514613028888042754926

`which has`, 55, `digits.`

[2, 17, 257, 353, 169553, 7699649, 197231873, 47072139617, 531968664833]

`Here is a solution:`

139215721114587930349655983948825318049741491427918

`which has`, 51, `digits.`

[2, 17, 257, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833]

`Here is a solution:`

14185411537593805288234493163329455708020259788019214

`which has`, 53, `digits.`

[2, 17, 353, 35969, 169553, 7699649, 197231873, 47072139617, 1110623386241]

`Here is a solution:`

40678439538499830791933157748038994675025900641347662

`which has`, 53, `digits.`

[2, 17, 353, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

601619593783137470771528342578494620959524176203704138227534

`which has`, 60, `digits.`

[2, 17, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

61302139288344678997680178340526552468252479019464685971405582

`which has`, 62, `digits.`

[2, 257, 353, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833]

`Here is a solution:`

294555898398271368632163299215017521466538335598281326

`which has`, 54, `digits.`

[2, 353, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

1272920892281510099187123703188580765958419123168884361641539438

`which has`, 64, `digits.`

`There were`, 14, `solutions altogether.`

(9.1)

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

[43, 48, 50, 51, 51, 52, 53, 53, 54, 54, 55, 60, 62, 64] **(9.2)**

```
> MAX1 := max(SOLN_L);
```

MAX1 := 1272920892281510099187123703188580765958419123168884361641539438 **(9.3)**

```
> seq(PI3(2^j*MAX1), j = 1..11);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1303470993696266341567614672065106704341421182124937586320936384511 **(9.4)**

```
> length(2^10 * MAX1);
```

67 **(9.5)**

```
> no_solns||2 := 14 * 11;
```

no_solns2 := 154 **(11)**

(iii, false & true) $s = 9$ **has number of supports = 12. There were 14 primitive solutions, the least having 44 digits,**

the largest (having a highest-factor-of-5: 5^2) having 65 digits

```
>              p := 7:
             fac := 2:   ### 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:
 print(``); S_potential||p := LEVEL||LEV[p];
 print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
          count := 0:
         SOLN_L := []:
  print(``); print(____________________); print(``);
 for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = false
              and member(5, L_) = true
       and 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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873]$

`` `Here there are`, 12, `support primes.` ``

$[5, 17, 257, 353, 35969, 169553, 7699649, 197231873, 47072139617]$

`` `Here is a solution:` ``

$23532637370391888792401683367890335767738435$

`` `which has`, 44, `digits.` ``

$[5, 17, 257, 353, 35969, 169553, 7699649, 531968664833, 1110623386241]$

`Here is a solution:`

1497553787229943562330900385749442442363209064355

`which has`, 49, `digits.`

[5, 17, 257, 353, 35969, 169553, 197231873, 531968664833, 1110623386241]

`Here is a solution:`

38360883512171171761571953326415923844449308020835

`which has`, 50, `digits.`

[5, 17, 257, 353, 35969, 169553, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

9155360323107587767841168462525617963471640395747715

`which has`, 52, `digits.`

[5, 17, 257, 353, 35969, 7699649, 197231873, 531968664833, 1110623386241]

`Here is a solution:`

1742023664421185413898991635994556519866886873445555

`which has`, 52, `digits.`

[5, 17, 257, 353, 35969, 7699649, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

415758264120688015246385996775739219163485473577456595

`which has`, 54, `digits.`

[5, 17, 257, 353, 35969, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

10649937568290709783744332582569803786532572220106887315

`which has`, 56, `digits.`

[5, 17, 257, 353, 169553, 7699649, 197231873, 47072139617, 531968664833]

`Here is a solution:`

348039302786469825874139959872063295124353728569795

`which has`, 51, `digits.`

[5, 17, 257, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833]

`Here is a solution:`

35463528843984513220586232908323639270050649470048035 `which has`, 53, `digits.`

[5, 17, 353, 35969, 169553, 7699649, 197231873, 47072139617, 1110623386241]

`Here is a solution:`

101696098846249576979832894370097486687564751603369155

`which has`, 54, `digits.`

[5, 17, 353, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

1504048984457843676928820856446236552398810440509260345568835

`which has`, 61, `digits.`

[5, 17, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

153255348220861697494200445851316381170631197548661714928513955

`which has`, 63, `digits.`

[5, 257, 353, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833]

`Here is a solution:`

736389745995678421580408248037543803666345838995703315

`which has`, 54, `digits.`

[5, 353, 35969, 169553, 7699649, 197231873, 47072139617, 531968664833, 1110623386241]

`Here is a solution:`

3182302230703775247967809257971451914896047807922210904103848595

`which has`, 64, `digits.`

`There were`, 14, `solutions altogether.`

(10.1)

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

[44, 49, 50, 51, 52, 52, 53, 54, 54, 54, 56, 61, 63, 64] **(10.2)**

```
> MAX1 := max(SOLN_L);
```

MAX1 := 3182302230703775247967809257971451914896047807922210904103848595 **(10.3)**

```
> seq(PI3(5^j*MAX1), j = 1..2);
```

1, 31823022307037752479678092579714519148960478079222109041038485951 **(10.4)**

```
> length(5 * MAX1);
```

65 **(10.5)**

```
> no_solns||3 := 14 * 2;
```

$no\_solns3 := 28$ **(12)**

(iv, true & true) $s = 9$ **has number of supports = 12. There were 40 primitive solutions, the least having 37 digits,**

the largest (having a highest-factor-of-2 and 5: $2^{11}.5^{2}$**) having 63 digits**

```
>              p := 7:
            fac := 2:  ### 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:
 print(``); S_potential||p := LEVEL||LEV[p];
 print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
          count := 0:
         SOLN_L := []:
 print(``); print(____________________); print(``);
 for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = true
              and member(5, L_) = true
      and 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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873]$

`Here there are`, 12, `support primes.`

____

`There were`, 40, `solutions altogether.`

____

(11.1)

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

[37, 38, 40, 41, 41, 42, 42, 43, 43, 44, 44, 45, 45, 46, 47, 47, 47, 48, 48, 48, 49, 50, 50, 50, 50, 51, 51, 52, 52, 53, 53, 53, 54, 54, 55, 56, 56, 57, 57, 59] **(11.2)**

```
> MAX1 := max(SOLN_L);
```
$$MAX1 := 273290388761046476572895406182472945026401376977083068848 70 \quad (11.3)$$

```
> seq(PI3(2^j*MAX1), j = 1..11);
```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -27984935809131159201064489593085229570703501002453306250106879 **(11.4)**

```
> seq(PI3(2^j* 5 * MAX1), j = 1..11);
```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -13992467904565579600532244796542614785351750501226653125053439 9 **(11.5)**

```
> length(2^10 * 5 * MAX1);
```
63 **(11.6)**

```
> ifactor(2^10 * 5 * MAX1);
```
$$(2)^{11} (5)^2 (7) (257) (7699649) (47072139617) (197231873) (1110623386241) (531968664833) (35969) \quad (11.7)$$

```
> no_solns||4 := 40 * 11 * 2;
```
$$no\_solns4 := 880 \quad (13)$$

```
> TOTAL||9 := add(no_solns||i, i = 1..4);
```
TOTAL9 := 1066 **(14)**

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

```
> s := 10: Pow(7^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
```
12 **(15)**

(i, false & false) s = 10 has number of supports = 14. There were 25 primitive solutions, having between 63 and 85 digits

```
>               p := 7:
              fac := 2:   ### 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:
 print(``); S_potential||p := LEVEL||LEV[p];
 print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
          count := 0:
         SOLN_L := []:
  print(``); print(____________________); print(``);
 for L_ in choose(S_potential||p, LEV) do
                if member(2, L_) = false
               and member(5, L_) = false
      and 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_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369]

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

____

`There were`, 25, `solutions altogether.`

____

(12.1)

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

[63, 69, 70, 71, 72, 72, 73, 73, 75, 76, 77, 78, 79, 79, 80, 80, 81, 81, 82, 82, 82, 83, 83, 84, 85] **(12.2)**

```
> no_solns||1 := 25;
```

no_solns1 := 25 **(16)**

(ii, true & false) s = 10 has number of supports = 14. There were 72 primitive solutions, the least having 51 digits,

the largest (having a highest-factor-of-2: 2^{12}) having 87 digits

```
>              p := 7:
             fac := 2:  ### 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:
  print(``); S_potential||p := LEVEL||LEV[p];
  print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
           count := 0:
          SOLN_L := []:
   print(``); print(____________________); print(``);
  for L_ in choose(S_potential||p, LEV) do
                if member(2, L_) = true
               and member(5, L_) = false
        and 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(``);

## SOLN_L;
```

S_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369]

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

`There were`, 72, `solutions altogether.`

(13.1)

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

[51, 56, 57, 57, 57, 58, 59, 59, 60, 60, 60, 60, 61, 61, 61, 61, 61, 62, 63, 63, 65, 65, 66, 67, 67, 67, 68, 68, 68, 68, 68, 68, 69, 69, 69, 69, 70, 70, 70, 70, 70, 70, 71, 71, 71, 71, 71, 72, 72, 72, 72, 73, 73, 74, 74, 76, 76, 77, 77, 77, 78, 79, 79, 80, 80, 80, 81, 81, 82, 83, 83] **(13.2)**

```
> MIN1 := min(SOLN_L);
```

MIN1 := 269847745529404336009629788387550334542285774939982 **(13.3)**

```
> seq(PI3(2^j*MIN1), j = 1..12);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -55264818284422008014772180661770308514260126707708313 **(13.4)**

```
> length(2^11 * max(SOLN_L));
```

87 **(13.5)**

```
> no_solns||2 := 72 * 12;
```

no_solns2 := 864 **(17)**

(iii, false & true) **s = 10 has number of supports = 14. There were 72 primitive solutions, the least having 51 digits,**

the largest (having a highest-factor-of-5: 5^2) having 85 digits

```
>              p := 7:

            fac := 2:  ### 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:

print(``); S_potential||p := LEVEL||LEV[p];
```

```
print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);
          count := 0:
         SOLN_L := []:
 print(``); print(____________________); print(``);
for L_ in choose(S_potential||p, LEV) do
               if member(2, L_) = false
              and member(5, L_) = true
      and 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(``);
## SOLN_L;
```

S_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369]

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

`There were`, 72, `solutions altogether.`

(14.1)

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

[51, 56, 57, 58, 58, 58, 59, 59, 60, 60, 61, 61, 61, 61, 62, 62, 62, 62, 63, 63, 65, 66, 66, 67, 67, 67, 67, 68, 68, 68, 68, 69, 69, 69, 69, 70, 70, 70, 70, 70, 71, 71, 71, 71, 71, 71, 71, 72, 72, 72, 73, 73, 73, 74, 75, 75, 77, 77, 77, 78, 78, 78, 79, 80, 80, 80, 80, 81, 82, 82, 83, 84] (14.2)

```
> MAX1 := max(SOLN_L);
```

MAX1 := 233726602622628594178058212583951860743080866370734569333697231973731003719165360115 (14.3)

```
> seq(PI3(5^j * min(SOLN_L)), j = 1..2);
```

1, -3373096819117554200120372354844379181778572186749774 (14.4)

```
> length(5 * MAX1);
```

85 (14.5)

```
> no_solns||3 := 72 * 2;
```

no_solns3 := 144 (18)

(iv, true & true) s = 10 has number of supports = 14. There were 162 primitive solutions, the least having 44 digits,

the largest (having a highest-factor-of-2 and 5: $2^{12}.5^2$) having 86 digits

```
>               p := 7:

             fac := 2:  ### 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:

 print(``); S_potential||p := LEVEL||LEV[p];

 print(``); lprint(`Here there are`, nops(S_potential||p), `support primes.`);

           count := 0:

          SOLN_L := []:

  print(``); print(____________________); print(``);

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

                 if member(2, L_) = true

                and member(5, L_) = true

       and 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(``);

 ## SOLN_L;
```

S_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369]

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

____

`There were`, 162, `solutions altogether.`

____

(15.1)

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

[44, 45, 46, 47, 48, 48, 48, 49, 49, 50, 50, 51, 51, 51, 52, 52, 52, 53, 54, 54, 54, 54, 54, 54, 55, 55, 55, 55, 55, 56, 56, 56, 56, 56, 57, 57, 57, 57, 57, 58, 58, 58, 58, 58, 58, 58, 59, 59, 59, 59, 59, 59, 59, 59, 60, 60, 60, 60, 60, 60, 60, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 62, 62, 62, 62, 62, 62, 63, 63, 63, 63, 63, 63, 63, 64, 64, 64, 64, 64, 64, 64, 64, 64, 65, 65, 65, 65, 65, 65, 65, 66, 66, 66, 66, 66, 66, 66, 66, 67, 67, 67, 67, 67, 67, 67, 67, 68, 68, 68, 68, 69, 69, 69, 69, 69, 69, 69, 70, 70, 70, 70, 70, 70, 71, 71, 71, 71, 71, 71, 71, 72, 72, 72, 72, 72, 73, 73, 73, 73, 73, 73, 74, 74, 74, 75, 75, 75, 76, 77, 82] **(15.2)**

```
> MAX1 := max(SOLN_L);
```

MAX1 := 1818884067102168048078274027890675959090123473702214547343947330534871624273660390 **(15.3)**

```
> seq(PI3(2^j * min(SOLN_L)), j = 1..12);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, −158330201264433302708205890170631079896788643839 **(15.4)**

```
> seq(PI3(2^j* 5 * min(SOLN_L)), j = 1..12);
```

(15.5)

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -7916510063221665135410294508531553994839432191999 **(15.5)**

```
> length(2^11 * 5 * MAX1);
```

86 **(15.6)**

```
> ifactor(2^11 * 5 * MAX1);
```

$(2)^{12} (5)^2 (7) (1110623386241) (169553) (197231873) (7699649) (47072139617) (126575155274810369) (531968664833) (28667393)$ **(15.7)**

```
> no_solns||4 := 162 * 12 * 2;
```

$no\_solns4 := 3888$ **(19)**

```
> TOTAL||10 := add(no_solns||i, i = 1..4);
```

TOTAL10 := 4921 **(20)**

____________________[end of s = 10, start of s = 11:]

```
> s := 11: Pow(7^(2^(s - 2 + 1)) - 1, 2); ### the highest power of '2' in supported solution
```

13 **(21)**

(i, false & false) s = 11 has number of supports = 18. There were 1449 primitive solutions, having between 67 and 176 digits

```
>               p := 7:
              fac := 2:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
              LEV := 11:
    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 := []:
    print(``); print(____________________); print(``);
   for L_ in choose(S_potential||p, LEV) do
                   if member(2, L_) = false
                  and member(5, L_) = false
         and 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(``);
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369, 13313, 27510200220671351632047923 3, 1338330888777063359811677099009, 656929861401793262700329631944023570433]$

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

`There were`, 1449, `solutions altogether.`
```

(16.1)

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

67
176

(16.2)

```
> no_solns||1 := 1449;
```

$no\_solns1 := 1449$ (22)

(ii, true & false) **s = 11 has number of supports = 18. There were 2685 primitive solutions, the least having 55 digits,**

the largest (having a highest-factor-of-2: 2^{13}) having 172 digits

The least solution was: the 55-digit 3592483036232959925296201372803457603761450521775980366

```
>               p := 7:
              fac := 2:   ### THE ABOVE PRE-COMPUTED VALUE OF fac
              LEV := 11:
   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 := []:
   print(``); print(____________________); print(``);
  for L_ in choose(S_potential||p, LEV) do
                 if member(2, L_) = true
                and member(5, L_) = false
        and 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(``);

## SOLN_L;
```

S_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369, 13313, 2751020022067135163204 79233, 1338330888777063359811677099009, 656929861401793262700329631944023570433]

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

____

`There were`, 2685, `solutions altogether.`

____

(17.1)

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

55

169 (17.2)

```
> MIN1 := min(SOLN_L);
```

MIN1 := 3592483036232959925296201372803457603761450521775980366 (17.3)

```
> seq(PI3(2^j*MIN1), j = 1..13);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1471481051641020385401324082300296234500690133719441557 9135 (17.4)

```
> length(2^12 * max(SOLN_L));
```

172 (17.5)

```
> no_solns||2 := 2685 * 13;
```

no_solns2 := 34905 (23)

IMPORTANT VISUAL NOTE. One could not but notice the (freakish) coincidence of the 2685 primitive solutions in cases (ii) and (iii), and we categorically assert that this is truly a coincidence (we thoroughly tested this, the least witness of which is to note the differing least solutions in these cases, both having (coincidentally!) 55 digits

(iii, false & true) s = 11 has number of supports = 18. There were 2685 primitive solutions, the least having 55 digits,

the largest (having a highest-factor-of-5: 5^2) having 170 digits

The least solution was: the 55-digit 8981207590582399813240503432008644009403626304439950915

```
>            p := 7:
```

```
            fac := 2:  ### THE ABOVE PRE-COMPUTED VALUE OF fac

            LEV := 11:

  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 := []:

  print(``); print(____________________); print(``);

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

                if member(2, L_) = false

               and member(5, L_) = true

      and 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(``);

 ## SOLN_L;
```

S_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369, 13313, 275102002206713516320479233, 1338330888770633598116770990009, 656929861401793262700329631944023570433]

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

`There were`, 2685, `solutions altogether.`

(18.1)

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

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

55

169 **(18.2)**

```
> MIN1 := min(SOLN_L);
```

MIN1 := 898120759058239981324050343200864400940362630443995091 5 **(18.3)**

```
> MAX1 := max(SOLN_L);
```

MAX1 := **(18.4)**

6060427179773572652695726034307027808541671791858170934086285287192257271947715213649377052080978339274658 4\
653869669845055002731771703988748219876550874876214912705365 15

```
> seq(PI3(5^j * min(SOLN_L)), j = 1..2);
```

1, -4490603795291199906620251716004322004701813152219975457 4 **(18.5)**

```
> length(5 * MAX1);
```

170 **(18.6)**

```
> no_solns||3 := 2685 * 2;
```

$no\_solns3 := 5370$ (24)

(iv, true & true) s = 11 has number of supports = 18. There were 3799 primitive solutions, the least having 49 digits,

the largest (having a highest-factor-of-2 and 5: $2^{13}.5^{2}$) having 164 digits

```
>              p := 7:
             fac := 2:  ### THE ABOVE PRE-COMPUTED VALUE OF fac
             LEV := 11:
  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 := []:
  print(``); print(____________________); print(``);
  for L_ in choose(S_potential||p, LEV) do
                if member(2, L_) = true
               and member(5, L_) = true
       and 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(``);
  ## SOLN_L;
```

$S\_potential7 := [2, 5, 17, 169553, 353, 47072139617, 7699649, 531968664833, 35969, 1110623386241, 257, 197231873, 28667393, 126575155274810369, 13313, 2751020022067135163204479233, 13383308887770633598116770990009, 656929861401793262700329631944023570433]$

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

```
`There were`, 3799, `solutions altogether.`
```

(19.1)

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

49

160 (19.2)

```
> MIN1 := min(SOLN_L);
```

$MIN1 := 1029223617887402616676926277266411897786107038790$ (19.3)

```
> seq(PI3(2^j * MIN1), j = 1..13);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -4215699938866801117908690031683223133331894430883839 (19.4)

```
> seq(PI3(2^j* 5 * MIN1), j = 1..13);
```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -2107849969433400558954345015841611566665947215441919 (19.5)

```
> MAX1 := max(SOLN_L);
```

$MAX1 :=$ (19.6)

4877415402679749856206427848744739225387831281650680303436358058921818115742220852614691840952313228850700 9\
1044942111715270834971993539832169513695180145867111 0

```
> length(2^12 * 5 * MAX1);
```

164 (19.7)

```
> ifactor(2^12 * 5 * MIN1);
```

$(2)^{13} (5)^{2} (7) (17) (257) (353) (7699649) (169553) (531968664833) (13313) (28667393) (35969)$ (19.8)

```
> no_solns||4 := 3799 * 13 * 2;
```

$no\_solns4 := 98774$ (25)

```
> TOTAL||11 := add(no_solns||i, i = 1..4);
```

TOTAL11 := 140498 (26)

[end of s = 11]

- **COMPLETE TO HERE (Tuesday 31st March 2015)**