

Le dernier chiffre se duplique plus loin

Envoyée fin janvier 2009 à la liste [SeqFan](#), cette idée de suite (je traduis) :

« Le dernier chiffre $\langle d \rangle$ de $a(n)$ est présent dans $a(n+1+d)$ »

Trois règles de construction supplémentaires :

- a) $a(1) = 1$
- b) quand il n'y a aucune contrainte d'écriture pesant sur $a(n)$, prendre $a(n) = a(n-1)+1$
- c) quand il y a une ou plusieurs contraintes d'écriture sur $a(n)$, prendre le plus petit $a(n) > a(n-1)$ ne conduisant pas à une contradiction.

C'est tout.

En français ordinaire il s'agit de construire une suite S strictement croissante d'entiers dont le dernier chiffre $\langle d \rangle$ est dupliqué quelque part dans un autre entier de la suite, situé à une distance $\langle d \rangle$ du premier, vers la droite. (Quand $\langle d \rangle$ vaut zéro, c'est le terme suivant qui doit contenir $\langle 0 \rangle$).

Sans réponse de SeqFan, j'ai envoyé ce problème au groupe rec.puzzles de Google.

Duncan Booth a calculé immédiatement les cent premiers termes de S ; il a joint son programme informatique :

```
S = 1, 2, 10, 20, 102, 103, 104, 112, 113, 123, 124, 134, 135, 136, 137,
140, 204, 205, 215, 216, 226, 234, 237, 245, 250, 260, 406, 416, 417, 425,
427, 428, 429, 436, 446, 450, 470, 480, 507, 508, 568, 569, 579, 580, 590,
600, 607, 608, 618, 628, 629, 639, 649, 650, 670, 680, 708, 718, 728, 729,
739, 749, 759, 760, 770, 780, 800, 801, 802, 819, 829, 892, 893, 894, 902,
903, 913, 920, 940, 1039, 1093, 1094, 1095, 1096, 1103, 1104, 1114, 1115,
1135, 1139, 1146, 1147, 1148, 1150, 1205, 1206, 1207, 1216, 1217, 1279,
...
```

```
from itertools import islice
def isok(s, needs):
    s = str(s)
    for c in needs:
        if c not in s:
            return False
    return True
def genS():
    required = {}
    n = 1
    S = 1
    while 1:
        obdigits = required.pop(n, "")
        while not isok(S, obdigits):
            S += 1
        yield n, S, obdigits
        d = S % 10
```

```

constrain = n+d+1
d = str(d)
if constrain in required:
    required[constrain] += d
else:
    required[constrain] = d
n += 1
S += 1
if __name__=='__main__':
    sequence = genS()
    for n,S,ob in islice(sequence, 100):
        print n,S,ob
---
```

Le tableau suivant reprend les cent premiers termes de la suite S (2^e colonne) et les chiffres à dupliquer (3^e colonne) :

1	1	
2	2	
3	10	1
4	20	0
5	102	20
6	103	
7	104	
8	112	2
9	113	
10	123	3
11	124	2
12	134	4
13	135	3
14	136	3
15	137	
16	140	4
17	204	40
18	205	
19	215	5
20	216	
21	226	6
22	234	4
23	237	7
24	245	5
25	250	5
26	260	0
27	406	640
28	416	6
29	417	
30	425	5
31	427	7
32	428	
33	429	
34	436	6
35	446	6
36	450	5
37	470	70
38	480	0
39	507	70
40	508	

41	568	86
42	569	6
43	579	9
44	580	
45	590	0
46	600	0
47	607	70
48	608	
49	618	8
50	628	8
51	629	
52	639	9
53	649	9
54	650	
55	670	70
56	680	0
57	708	80
58	718	8
59	728	8
60	729	
61	739	9
62	749	9
63	759	9
64	760	
65	770	0
66	780	80
67	800	80
68	801	80
69	802	
70	819	91
71	829	9
72	892	92
73	893	9
74	894	
75	902	2
76	903	
77	913	3
78	920	2
79	940	40
80	1039	930
81	1093	93
82	1094	
83	1095	
84	1096	
85	1103	3
86	1104	
87	1114	4
88	1115	
89	1135	53
90	1139	9
91	1146	64
92	1147	4
93	1148	
94	1150	5
95	1205	50
96	1206	
97	1207	

98	1216	6
99	1217	
100	1279	97

...

Merci à **Duncan Booth**, **Hans Havermann** et **Paolo Lava** pour leurs calculs, conseils et remarques.

Back to main page, [here](#).