

Savoir compter jusqu'à 1

Année 2018 – 2019

Castex Thibaut, Manresa Louis, Claret Gaëtan, Mariotte Margot, Dezcazeaux Kévin, classe de 4ème

Encadrés par Mr Lefrancois, Mr Jubert, Mme Poupin, Mme Pons et Mme Decamps

Établissements : Collège Nelson Mandela (Noé, 31), Collège André Abbal (Carbonne, 31)

Chercheur : Yohann Genzmer, Institut de Mathématiques de Toulouse

1. Présentation du sujet

On écrit les nombres de 1 à N.

Quel est le plus petit entier N supérieur ou égal à 2 tel que le nombre de fois que l'on écrit 1(**un**) pour ce faire est exactement N ?

Exemple :

N=10 donc on compte combien il y a de **UN** entre 1 et 10.

1;2;3;4;5;6;7;8;9;10

On remarque que il y a 2 fois le chiffre **1**. Donc 10 ne répond pas au problème **(1)**.

2. Annnonce des conjectures et résultats obtenus

- . Nombres de **un** de 0 à 100,200,300,400,500,600,700,800,900 et 1000
- . On a conjecturé que le nombre de un entre 0 et 10^n est $n \times 10^{n-1} + 1$.
- . Résolution du problème grâce à des logiciels informatiques scratch et python

3. Texte de l'article

a) première recherche

On a commencé par écrire les nombres de 1 à 100 sur le cahier on a compté 21 **un** (cela ne résoud pas le problème).

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
95 96 97 98 99 100

De la même manière on a écrit les nombres de 0 à 200 on a compté 140 **un** (Cela ne résoud toujours pas le

problème).

Après avoir compté les nombres de **un** de 0 à 200, on a essayé de trouver combien il y avait de **un** dans les nombres compris entre 0 et 300 mais on ne les a pas écrit on a commencé à additionner les résultats précédents [\(2\)](#).

De 0 à 99 il y a 20 **un** donc de 200 à 299 il y en a 20 **un** (comme de 0 à 99)

On additionne $140 + 20 = 160$

donc on a compté 160 **un**

b) Nombre de **un** de 0 à 10^n

Le nombre de **un** de 0 à 10^n est donné par la formule $n \times 10^{n-1} + 1$

Exemple :

Entre 0 et $1000=10^3$ il y a 301 **un** car $3 \times 10^{3-1} + 1 = 301$.

Explication de la formule :

On a trouvé la formule en comptant les **un** jusqu'aux premières puissances de 10.

Pour $1000=10^3$:

21 est le nombre de **un** de 0 à 100,

20×9 est le nombre de **un** des unités et des dizaines entre 101 et 999. (car il y a 20 **un** de 0 à 99 et 9 centaines)

99 est le nombre de **un** des chiffres des centaines de 100 à 199 .

1 est le nouveau **un** de 1 000.

on additionne est on trouve : $21 + 20 \times 9 + 99 + 1 = 301 = 3 \times 10^{3-1} + 1$.

Pour $10\ 000=10^4$:

301 est le nombre de **un** de 0 à $1\ 000=10^3$,

300×9 est le nombre de **un** des unités et des dizaines entre 1 001 et 9 999.

999 est le nombre de **un** du chiffre des milliers de 1 000 à 1 999 .

1 est le nouveau **un** de 1 000.

On additionne et on trouve : $301 + 300 \times 9 + 999 + 1 = 4\ 001 = 4 \times 10^{4-1} + 1$.

Pour $100\ 000=10^5$:

4 001 est le nombre de **un** de 0 à $10\ 000=10^4$,

4000×9 est le nombre de **un** des unités et des dizaines entre 10 001 et 99 999.

999 c'est le nombre de **un** de 10 000 à 19 999 .

1 est le nouveau **un** de 10 000.

On additionne et on trouve : $4\ 001 + 4\ 000 \times 9 + 9\ 999 + 1 = 50\ 001 = 5 \times 10^{5-1} + 1$.

Aux vus des résultats on a conjecturé que le nombre de **un** de 0 à 10^n est $n \times 10^{(n-1)} + 1$

De 0 jusqu'à 10^{10} on trouve 10 000 000 001 donc cela se rapproche fortement d'un résultat au problème [\(3\)](#).

Nous avons donc essayé de reprendre nos calculs pour savoir s'il existait un nombre qui répondrait au problème [\(4\)](#) et qui est inférieur à 10 000 000 001.

c) Avec des programmes informatiques

Avec Scratch :

Voici le programme que nous avons écrit, il compte le nombre de **un** en utilisant des variables et le reste de différentes divisions par 10.

```

définir chiffre des unités
si 1 = N modulo 10 alors
    ajouter à NOMBRE DE FOIS 1 1
sinon
    ajouter à NOMBRE DE FOIS 1 0

définir chiffre des dizaines
si 1 = (N modulo 100 - N modulo 10) / 10 alors
    ajouter à NOMBRE DE FOIS 1 1

définir chiffre des centaines
si 1 = (N modulo 1000 - N modulo 100) / 100 alors
    ajouter à NOMBRE DE FOIS 1 1

définir chiffres des milliers
si 1 = (N modulo 10000 - N modulo 1000) / 1000 alors
    ajouter à NOMBRE DE FOIS 1 1

définir CHIFFRE DES DIX MILIERS
si 1 = (N modulo 100000 - N modulo 10000) / 10000 alors
    ajouter à NOMBRE DE FOIS 1 1

définir Chiffre des centaines de milliers
si 1 = (N modulo 1000000 - N modulo 100000) / 100000 alors
    ajouter à NOMBRE DE FOIS 1 1

définir chiffre des millions
si 1 = (N modulo 10000000 - N modulo 1000000) / 1000000 alors
    ajouter à NOMBRE DE FOIS 1 1

quand est cliqué
supprimer l'élément tout de la liste liste des r
mettre NOMBRE DE FOIS 1 à 0
mettre N à 0
répéter jusqu'à 2600001 < N
    chiffre des unités
    chiffre des dizaines
    chiffre des centaines
    chiffres des milliers
    CHIFFRE DES DIX MILIERS
    Chiffre des centaines de milliers
    chiffre des millions
    chiffre des dizaines de millions
    chiffre des centaines de millions
    Chiffre des unites de milliards
    chiffre des dizaines de milliards
    chiffre des centaines de milliards
    chiffre des millions de milliards
    si N = NOMBRE DE FOIS 1 alors
        ajouter N à liste des nombres
    ajouter à N 1

quand espace est pressé
mettre NOMBRE DE FOIS 1 à 0
demander donner un nombre et attendre
mettre N à réponse
    chiffre des unités
    chiffre des dizaines
    chiffre des centaines
    chiffres des milliers
    CHIFFRE DES DIX MILIERS
    Chiffre des centaines de milliers
    chiffre des millions
    chiffre des dizaines de millions
    chiffre des centaines de millions
    Chiffre des unites de milliards
    chiffre des dizaines de milliards
    chiffre des centaines de milliards
    chiffre des millions de milliards
    
```

Avec Python :

```
nn = 0
max = 100
for n in range(0, max + 1):
    nn += str(n).count("1")
    if n == nn:
        print(nn)
        break
```

Traduction du programme :

CRÉATION de la variable *NN* et ASSIGNATION de la valeur 0
 POUR CHAQUE valeur de la liste de 0 à 101
 ALORS
 ASSIGNATION de *NN* à la valeur de l'ancienne valeur de *NN* + le nombre de fois que le symbole 1
 apparaît au rang *n*
 SI *N* est égal à *NN* ALORS
 On affiche la valeur de *NN*
 On stoppe la boucle

Cela a permis de trouver beaucoup de résultats.

```
1 nn = 0
2 max = 10000000
3 for n in range(0, max + 1):
4     nn += str(n).count("1")
5     if(n == nn):
6         print(nn)
7
```

4. Conclusion

Nous avons finalement trouvé la réponse que l'on cherchait grâce aux logiciels :
 lorsque l'on écrit tous les nombres de 0 à 199 981 on écrit 199 981 fois le nombre **un**.

Nous pensons qu'il n'y a qu'un nombre fini de solutions, mais nous ne l'avons pas démontré.

Voici les 24 premières solutions :

199 981	1 599 981	2 600 000
199 982	1 599 982	2 600 001
199 983	1 599 983	
199 984	1 599 984	
199 985	1 599 985	
199 986	1 599 986	
199 987	1 599 987	
199 988	1 599 988	
199 989	1 599 989	
199 990	1 599 990	

200 000		
200 001		

Notes d'édition

(1) 10 ne répond pas à la question posée, ou bien 10 n'est pas une solution du problème

(2) On a utilisé les résultats des calculs précédents

(3) se rapproche fortement d'une solution du problème

(4) serait solution du problème