Cet article est rédigé par des élèves. Il peut comporter des oublis et imperfections, autant que possible signalés par nos relecteurs dans les notes d'édition.

Fin des puissances de 2

Année 2024 – 2025

Robin Hubert, Eliott Fougeron Courgeon, Tom Boirivant, Gaspard Barré, Noa Bernard, Théodore Desmarchelier, Alexandre Gaboriau, Milo Porterfield,
élèves de première et terminale

Établissement(s) : Lycée Léonce Vieljeux, Lycée René Josué Valin - La Rochelle

Enseignant·e(s) : Rachel Biton, Pierre Vederine, Jean-Matthieu Bernat

Chercheur·Chercheuse(s) : Ospel Cyrille, Université de La Rochelle

1. Introduction

1.1. Présentation du sujet

L'objectif de ce problème est de déterminer les 3 derniers chiffres du résultat d'une puissance de 2 uniquement à partir de son exposant .
Á quelles conditions existe-t-il une puissance de 2 finissant par un nombre donné ?

1.2. Résultats

Nos recherches nous ont permis de déterminer un algorithme permettant de déterminer les derniers chiffres du résultat d'une puissance de 2 uniquement à partir de son exposant.

2. Conjectures

2.1. Tableau donnant les trois derniers chiffres des premières puissances de 2 :

Le tableau ci dessous indique les trois derniers chiffres (chiffre des centaines, chiffre des dizaines et chiffre des unités) de chaque puissance de 2, en commençant par 2^1 jusque 2^{20} dans la première colonne, puis dans la deuxième colonne les trois derniers chiffres de 2^{21} jusque 2^{40} , et dans la troisième colonne de 2^{41} à 2^{60} ...

002	152	552	952	352	752
004	304	104	904	704	504
008	608	208	808	408	008
016	216	416	616	816	

032	432	832	232	632	
064	864	664	464	264	
128	728	328	928	528	
256	456	656	856	056	
512	912	312	712	112	
024	824	624	424	224	
048	648	248	848	448	
096	296	496	696	896	
192	592	992	392	792	
384	184	984	784	584	
768	368	968	568	168	
536	736	936	136	336	
072	472	872	272	672	
144	944	744	544	344	
288	888	488	088	688	
576	776	976	176	376	

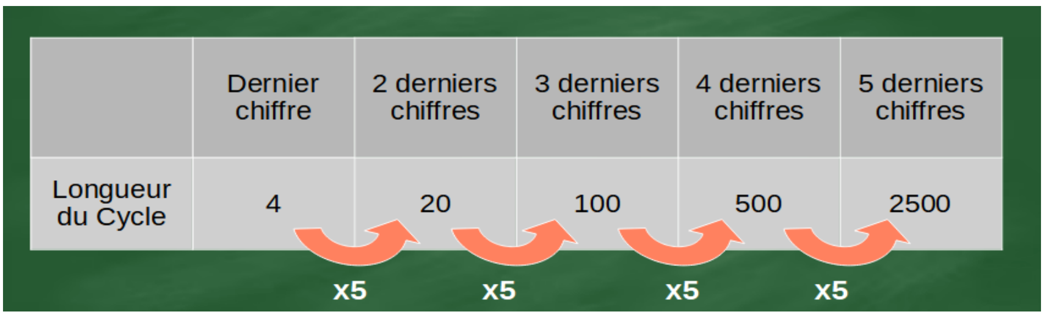
Conjecture n°1 :
Le résultat de 2^n a un chiffre des unités qui suit un cycle de longueur 4 (2-4-8-6)
$2^n = \overline{\dots a}$ alors $a=2$ puis $a=4$ puis $a=8$ puis $a=6$

Conjecture n°2 : Le résultat de 2^n se termine par deux chiffres qui forment un nombre qui suit un cycle de longueur 20 (04-08-16-32,,,)
$2^n = \overline{\dots ba}$ alors ba suit un cycle de longueur 20 où $ba=$

04 – 08 – 16 – 32 – 64 – 28 – 56 – 12 – 24 – 48 – 96 – 92 – 84 – 68 – 36 – 72 – 44 – 88 – 76 – 52

Conjecture n°3 : Le résultat de 2^n se termine par trois chiffres qui forment un nombre qui suit un cycle de longueur 100. (1)

2.2. Conjectures sur la longueur du cycle :



Conjecture n°4 : la longueur du cycle est multiplié par 5 lorsque l'on prend en compte un chiffre de plus dans l'écriture du nombre.

3. Un programme python

Afin de répondre à une des questions, nous avons réalisé un programme Python permettant de rechercher les exposants dont le résultat se finit par une séquence de chiffres donnée dans un certain échantillon :

```python
import sys
r = 10000
f = 0
g = 0
e = int(input("Séquence recherchée:"))
d = len(str(e))


for n in range(1,r):
    t=2**n
    a=str(t)
    b=a[-d:]
    if int(b)==e:
        if len(a) == d:
            print("Exposant:", n)
            print(b)

        else:
            c=a[len(a)-(d+1):]
            print("Exposant:", n)
            print(d+1,"derniers chiffres:", c)
        g = 1
        f = f+1

if g==1:
    print(f, "exposant(s) trouvé(s) sur l'échantillon de",r,",",f/r*100,"% des puissances de termine par",e)
else:
    print("rien de trouvé")


```

Explication du code **Se reporter à « *Annexe* » *page 10***

4. Démonstration des conjectures

Pour démontrer nos conjectures, nous allons démontrer les propriétés suivantes :

Propriété n°1 :
Pour $n \geqslant 1$, Le résultat de 2^n a un chiffre des unités qui suit un cycle de longueur 4 (le cycle est 2-4-8-6).
Si $2^n = \overline{\dots a}$ alors $a=2$ puis $a=4$ puis $a=8$ puis $a=6$.

Plus précisément :
si $n \equiv 1 [4]$ alors le chiffre des unités de 2^n est 2.
si $n \equiv 2 [4]$ alors le chiffre des unités de 2^n est 4.
si $n \equiv 3 [4]$ alors le chiffre des unités de 2^n est 8.
si $n \equiv 0 [4]$ alors le chiffre des unités de 2^n est 6.

Propriété n°2 :

Pour $n \geqslant 2$, le résultat de 2^n se termine par deux chiffres qui forment un nombre qui suit un cycle de longueur 20 .
(le cycle est 04-08-16-32-64-28-56-12-24-48-96-92-84-68-36-72-44-88-76-52)
Si $2^n = \overline{\dots ba}$ alors $ba = 04$ puis $ba = 08$ .....

<u>Propriété n°3</u> :
Pour $n \geqslant 3$, le résultat de 2^n se termine par trois chiffres qui forment un nombre qui suit un cycle de longueur 100.

<u>Propriété n°4</u> :
Pour $n \geqslant 4$, le résultat de 2^n se termine par quatre chiffres qui forment un nombre qui suit un cycle de longueur 500

<u>Propriété n°5</u> :
Pour $n \geqslant m$, le résultat de 2^n se termine par m chiffres qui forment un nombre qui suit un cycle de longueur $4 \times 5^{m-1}$
Donc la longueur de ce cycle est multiplié par 5 lorsque l'on prend en compte un chiffre de plus dans l'écriture du nombre.

4.1. Démonstration de la propriété n°1

Étape 1 : Montrons que $\forall\, k \in \mathbb{N}$: $2^{1+4k} \equiv 2[10]$.

$2^{1+4k} \equiv 2[10]$

$\Leftrightarrow \quad 2^{1+4k} - 2 \equiv 0[10]$

$\Leftrightarrow \quad 2 \times (2^{4k} - 1) \equiv 0[10]$

$\Leftrightarrow$ il existe un entier q tel que $2(2^{4k} - 1) = 10q$

$\Leftrightarrow$ il existe un entier q tel que $2(2^{4k} - 1) = 2 \times 5 \times q$

$\Leftrightarrow$ il existe un entier q tel que $2^{4k} - 1 = 5q$

$\Leftrightarrow \quad 2^{4k} - 1 \equiv 0[5]$

$\Leftrightarrow \quad 2^{4k} \equiv 1[5]$

$\Leftrightarrow \quad 16^k \equiv 1[5]$

Or $16 \equiv 1[5]$ donc $16^k \equiv 1^k$ [5]

Ainsi $\forall\, k \in \mathbb{N}$: $2^{1+4k} \equiv 2[10]$

Étape 2 : $\forall\, k \in \mathbb{N}$, $2^{1+4k} \equiv 2[10]$

Donc $2 \times 2^{1+4k} \equiv 2 \times 2[10]$ donc $2^{2+4k} \equiv 4[10]$

donc $2\times 2^{2+4k}\equiv 2\times 4[10]$ donc $2^{3+4k}\equiv 8[10]$

donc $2\times 2^{3+4k}\equiv 2\times 8[10]$ donc $2^{4+4k}\equiv 16[10]\equiv 6[10]$ soit $2^{4(k+1)}\equiv 6[10]$

Exemple :

- $2^{27}=2^{4\times 6+3}$ donc 2^{27} a pour chiffre des unités le chiffre 8.
  En effet : $2^{27}=134217728$
- $2^{53}=2^{4\times 13+1}$ donc 2^{53} a pour chiffre des unités le chiffre 2.
  En effet : $2^{53}=9\,007\ldots 992$

4.2. Démonstration de la propriété n°2

étape 1 : Montrons que $\forall\ k\in\mathbb{N}\ :\ 2^{2+20k}\equiv 2^2[100]$

$2^{2+20k}\equiv 2^2[100]$

$\Leftrightarrow\quad 2^{2+20k}-2^2\equiv 0[100]$

$\Leftrightarrow\quad 2^2\times 2^{20k}-2^2\equiv 0[100]$

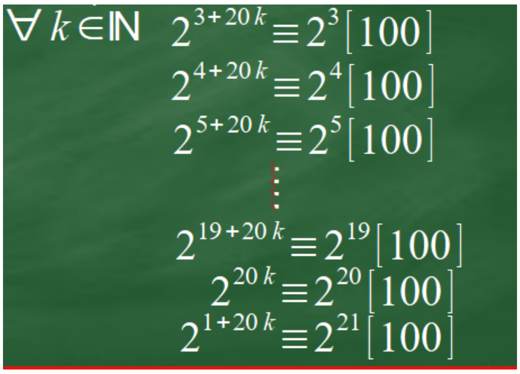
$\Leftrightarrow\quad 2^2\times(2^{20k}-1)\equiv 0[100]$

$\Leftrightarrow$ il existe un entier q tel que $2^2(2^{20k}-1)=100q$

$\Leftrightarrow$ il existe un entier q tel que $2^2(2^{20k}-1)=2^2\times 25\times q$

$\Leftrightarrow$ il existe un entier q tel que $2^{20k}-1=25q$

$\Leftrightarrow\quad 2^{20k}-1\equiv 0[25]$
$2^{20k}\equiv 1[25]$ (2)



Or $2^{20}\equiv 1[25]$ donc $(2^{20})^k\equiv 1^k[25]$ donc $2^{20k}\equiv 1[25]$

Ainsi $\forall\ k\in\mathbb{N}\ ::\ 2^{2+20k}\equiv 2^2[100]$

étape 2 : $2^{2+20k}\equiv 2^2[100]$

donc $2\times 2^{2+20k}\equiv 2\times 2^2[100]$ $2^{3+20k}\equiv 2^3[100]$
donc $2\times 2^{2+20k}\equiv 2\times 2^2[100]$

4.3. Démonstration des propriétés n°3 et n°4

On montre de façon analogue que $\forall\ k\in\mathbb{N}\ :\ 2^{3+100k}\equiv 2^3[1000]$

et que $\forall\ k\in\mathbb{N}\ :\ 2^{4+500k}\equiv 2^4[10000]$ (3)

4.4. Démonstration de la propriété n°5

Cas général :

On veut démontrer que pour tout entier naturel n et pour tout entier naturel k :

$2^{n+4k\times 5^{n-1}} \equiv 2^n \left[10^n\right]$

Étape 1 : On montre que $2^{n+4k\times 5^{n-1}} \equiv 2^n \left[10^n\right] \;\Leftrightarrow\; \left(16^{5^{n-1}}\right)^k \equiv 1 \left[5^n\right]$

Démonstration :

Soit k un entier naturel, et n un entier naturel n≥1:

$$2^{n+4k\times 5^{n-1}} \equiv 2^n \left[10^n\right]$$

$$\Leftrightarrow \quad 2^{n+4k\times 5^{n-1}} - 2^n \equiv 0 \left[10^n\right]$$

$$\Leftrightarrow \quad 2^n \times \left(2^{4k\times 5^{n-1}} - 1\right) \equiv 0 \left[10^n\right]$$

$\Leftrightarrow$ il existe un entier q tel que $2^n \times \left(2^{4k\times 5^{n-1}} - 1\right) = 10^n \times q$

$\Leftrightarrow$ il existe un entier q tel que $2^n \times \left(2^{4k\times 5^{n-1}} - 1\right) = 2^n \times 5^n \times q$

$\Leftrightarrow$ il existe un entier q tel que $\left(2^{4k\times 5^{n-1}} - 1\right) = 5^n \times q$

$$\Leftrightarrow \quad \left(2^{4k\times 5^{n-1}} - 1\right) \equiv 0 \left[5^n\right]$$

$$\Leftrightarrow \quad 16^{k\times 5^{n-1}} - 1 \equiv \left[5^n\right]$$

$$\Leftrightarrow \quad 16^{k\times 5^{n-1}} \equiv 1 \left[5^n\right]$$

$$\Leftrightarrow \quad \left(16^{5^{n-1}}\right)^k \equiv 1 \left[5^n\right]$$

Étape 2 : Il suffit de montrer que $16^{5^{n-1}} \equiv 1 \left[5^n\right]$

car si $16^{5^{n-1}} \equiv 1 \left[5^n\right]$ alors $\left(16^{5^{n-1}}\right)^k \equiv 1^k \left[5^n\right]$ et donc $\left(16^{5^{n-1}}\right)^k \equiv 1 \left[5^n\right]$

Étape 3 : Démonstration de $16^{5^{n-1}} \equiv 1 \left[5^n\right]$ en utilisant le binôme de Newton : (4)

$$16^{5^{n-1}} \equiv 1 \left[5^n\right]$$

$$\Leftrightarrow \quad 16^{5^{n-1}} - 1 \equiv 0 \left[5^n\right]$$

$\Leftrightarrow$ il existe un entier p tel que $16^{5^{n-1}} - 1 = 5^n \times p$

Or $16^{5^{n-1}} = (15+1)^{5^{n-1}}$

D'après la formule du binôme de Newton :

$$(15+1)^{5^{n-1}} = 15^{5^{n-1}} + \binom{5^{n-1}}{1} 15^{5^{n-1}-1} \times 1 + \binom{5^{n-1}}{2} 15^{5^{n-1}-2} \times 1^2 + \ldots\ldots + \binom{5^{n-1}}{5^{n-1}-1} \times 15^1 + 1$$ (5)

Il reste à montrer que :

a) $\binom{5^{n-1}}{M}\times 15^{5^{n-1}-M}$ est un multiple de 5^n pour M entier compris entre 1 et $5^{n-1}-1$,

b) 5^n divise $15^{5^{n-1}}$

Une fois a) et b) démontrés, on peut en déduire que $16^{5^{n-1}}\equiv 1\left[5^n\right]$

Démonstration du a) pour M entier compris entre 1 et $5^{n-1}-1$,

on montre que $\binom{5^{n-1}}{M}\times 15^{5^{n-1}-M}$ est un multiple de 5^n

(pour M entier compris entre 1 et $5^{n-1}-1$)

Or $\binom{5^{n-1}}{M}=\left(\dfrac{5^{n-1}!}{M!\left(5^{n-1}-M\right)!}\right)$ est un entier

$\binom{5^{n-1}}{M}=\left(\dfrac{5^{n-1}!}{M!\left(5^{n-1}-M\right)!}\right)=\dfrac{5^{n-1}\times\left(5^{n-1}-1\right)\times\left(5^{n-1}-2\right)\times\ldots\times\left(5^{n-1}-\mathrm{M}+1\right)}{\mathrm{M}!}$ est un entier.

<u>Si M<5,</u> alors $\mathrm{M}!$ ne divise pas 5^{n-1} donc $\dfrac{\left(5^{n-1}-1\right)\times\left(5^{n-1}-2\right)\times\ldots\times\left(5^{n-1}-\mathrm{M}+1\right)}{\mathrm{M}!}$ est un entier.

$\binom{5^{n-1}}{M}=5^{n-1}\times\mathrm{Q}$ avec Q un entier.

5^{n-1} divise $\binom{5^{n-1}}{M}$ donc $5^{n-1}\times 5$ divise $\binom{5^{n-1}}{M}\times 5$

$\binom{5^{n-1}}{M}\times 15^{5^{n-1}-M}$ est un multiple de 5^n

Si M=5 : (6)

$\binom{5^{n-1}}{M}=\left(\dfrac{5^{n-1}!}{M!\left(5^{n-1}-M\right)!}\right)=\dfrac{5^{n-1}\times\left(5^{n-1}-1\right)\times\left(5^{n-1}-2\right)\times\ldots\times\left(5^{n-1}-\mathrm{M}+1\right)}{\mathrm{M}!}=\dfrac{5^{n-2}\times\left(5^{n-1}-1\right)\times\left(5^{n-1}-2\right)\times\ldots\times\left(5^{n-1}-\mathrm{M}+1\right)}{(\mathrm{M}\text{-}1)!}$

est un entier , et $\left(\mathrm{M}-1\right)!$ ne divise pas 5^{n-2}

donc 5^{n-2} divise $\binom{5^{n-1}}{M}$ donc $5^{n-2}\times 5^2$ divise $\binom{5^{n-1}}{M}\times 5^2$

$\binom{5^{n-1}}{M}\times 15^{5^{n-1}-M}$ est un multiple de 5^n

Et sur le même principe au peut démontrer que $\binom{5^{n-1}}{M}\times 15^{5^{n-1}-M}$ est un multiple de 5^n (7)

(pour M entier compris entre 1 et $5^{n-1}-1$)

Démonstration du b) Il faut montrer aussi que 5^n divise $15^{5^{n-1}}$

Il faut donc montrer que $5^n \mid 3^{5^{n-1}} \times 5^{5^{n-1}}$

Pour cela il suffit de montrer par récurrence que $5^{n-1} \geqslant n$

Initialisation pour $n=1$:
$5^{1-1}=5^0=1$ et $1 \geqslant 1$ donc la propriété est vraie au rang 1.

Hérédité : Soit n fixé. On suppose que $5^{n-1} \geqslant n$

$5^{n-1} \geqslant 1$ donc $5 \times 5^{n-1} \geqslant 5 \times n$
donc $5^n \geqslant 5n$

Et $5n \geqslant n+1 \quad \Leftrightarrow \quad n \geqslant \frac{1}{4}$
Or n est un entier naturel non nul, donc $5n \geqslant n+1$.
Ainsi $5^n \geqslant 5n \geqslant n+1$
$5^n \geqslant n+1$. La propriété est donc vraie au rang $n+1$.

Conclusion : la propriété est vraie au rang 1 et elle est héréditaire, donc pour tout entier naturel non nul n : $5^{n-1} \geqslant n$

Étape 3 autre méthode : Démonstration de $16^{5^{n-1}} \equiv 1 [5^n]$ en utilisant la fonction indicatrice d'Euler :

$16=2^4$ donc $16^{5^{n-1}}=(2^4)^{5^{n-1}}=2^{4\times 5^{n-1}}$

$16^{5^{n-1}} \equiv 1 [5^n] \Leftrightarrow 2^{4\times 5^{n-1}} \equiv 1 [5^n]$.
Nous allons donc montrer que $2^{4\times 5^{n-1}} \equiv 1 [5^n]$

Soit ϕ la fonction indicatrice d'Euler.
$\phi(p)$ est le nombre d'entier compris entre 1 et p, premier avec p .

Propriétés de l'indicatrice d'Euler : (8)

- $\phi(p)=p-1$ si p est un nombre premier.
- $\phi(p^n)=p^n-p^{n-1}$ si p est un nombre premier.
- si $pgcd(m,a)=1$ alors $a^{\phi(m)} \equiv 1 [m]$

Posons $a=2$ et $m=5^n$. $pgcd(2,5^n)=1$ donc $pgcd(a,m)=1$
et $\phi(m)=\phi(5^n)=5^n-5^{n-1}=5^{n-1}(5-1)=5^{n-1}\times 4$

$pgcd(a,m)=1$ donc $a^{\phi(m)} \equiv 1 [m]$. Ainsi $2^{4\times 5^{n-1}} \equiv 1 [5^n]$

Nous avons donc montrer que $2^{4\times 5^{n-1}} \equiv 1 [5^n]$

Ainsi pour tout entier naturel n : $16^{5^{n-1}} \equiv 1 [5^n]$

5. Conclusion

On a donc démontré que pour tout entier naturel n et pour tout entier naturel k :
$2^{n+4k\times 5^{n-1}} \equiv 2^n \left[10^n\right]$.
Ainsi le résultat de 2^n se termine par m chiffres qui forment un nombre qui suit un cycle de longueur $4\times 5^{m-1}$
Donc la longueur de ce cycle est multiplié par 5 lorsque l'on prend en compte un chiffre de plus dans l'écriture du nombre.

Notes d'édition

(1) Ce cycle commence à $8=2^3$

(2) En effet : 2^20 = 1048576

(3) On doit également exploiter ici la connaissance de l'écriture de
2^100 = 1267650600228229401496703205376=1 [125], et de
2^500 = 32733906078961418700131896968275991522166420460430647894832913680961337964046745548832700923259041571508866841275600710092172565458853930533285275893 76=1 [625]. La limite à cette méthode apparaît rapidement.

(4) La démonstration proposée pour cette étape est incomplète et n'est valable que pour n≥2. L'édition laisse à titre d'exercice l'établissement de la preuve basée sur la formule de récurrence 16^{5^n} = (16^{5^{n-1}})^5, et la formule du binôme de Newton pour n=5, à savoir $(a+b)^5=a^5 + 5ab^4 + 10a^2b^3 + 10a^3b^2 + 5a^4b + b^5$. On tiendra compte du fait que la propriété 1 établit la propriété pour n=1 et peut donc servir d'initialisation.

(5) On pourrait aussi obtenir une expression plus simple et inversant 1 et 15, et un terme générique de somme qui prend la forme : \sum_{M=0}^{5^{n-1}} \begin{pmatrix}5^{n-1}\\M\end{pmatrix}15^M.

(6) Il y a ici une simplification par 5 qui n'est valable que si n≥2.

(7) La généralisation n'est ici pas évidente et il manque un argument pour la guider et convaincre qu'elle est valide.

(8) Si la première propriété est assez simple à établir, les deux suivantes le sont moins, et il aurait été intéressant de présenter au moins une esquisse de preuve pour ces notions hors programme du lycée.

Annexe

Programme python
Afin de répondre à une des questions, nous avons réalisé un programme Python permettant de rechercher les exposants dont le résultat se finit par une séquence de chiffres donnée dans un certain échantillon :

```
1  import sys
2  r = 10000
3  f = 0
4  g = 0
5  e = int(input("Séquence recherchée:"))
6  d = len(str(e))
7
8
9  for n in range(1,r):
10     t=2**n
11     a=str(t)
12     b=a[-d:]
13     if int(b)==e:
14         if len(a) == d:
15             print("Exposant:", n)
16             print(b)
17
18         else:
19             c=a[len(a)-(d+1):]
20             print("Exposant:", n)
21             print(d+1,"derniers chiffres:", c)
22         g = 1
23         f = f+1
24
25 if g==1:
26     print(f, "exposant(s) trouvé(s) sur l'échantillon de",r,",",f/r*100,"% des puissances de termine par",e)
27 else:
28     print("rien de trouvé")
29
30
```

Note d'édition : *« L'usage de « fonction » pour if, for, etc est inexact.*
Ce sont des structures de contrôle (condition, boucle). »

Explication du code

Ligne 1 : importation d'une fonction qui va nous permettre d'améliorer notre programme. Ici « sys » qui va nous permettre d'augmenter le nombre de répétions possibles de la boucle « for » sans qu'il y ait de problèmes lors de l'exécution du programme.

Ligne 2 : création d'une variable « r » qui sera la longueur de la séquence.

Ligne 3-4 : création de variables numériques de stockage de données (« f », « g »).

Ligne 5 : demande à l'utilisateur une valeur numérique stocké dans une variable « e » servant à connaître les derniers chiffres voulant être trouvés par le résultat de 2**n.

Ligne 6 : introduit dans une nouvelle variable la séquence recherchée en la changeant en une chaîne de caractères par la fonction « str() » puis le ramenant à un nombre « int(e) » par la fonction len() en calculant le nombre de caractères dans la séquence.

Ligne 9 : implantation d'une boucle « FOR » pour répéter les instructions demandées de 1 à « r » fois pour n.

Ligne 10 : création d'une variable « t » qui permet le calcule : 2**n.

Ligne 11 : transforme « t » en une chaîne de caractères dans la variable a.

Ligne 12 : récupère de « a » les dernier caractères (chiffres) de la chaîne en faisant notre coupage par la longueur de la séquence recherché démarrant depuis la fin moins « d » pour prendre ce qui reste de la chaîne, stockée dans la variable b.

Ligne 13 : introduit une fonction « IF » pour déterminer lorsque le nombre entier du coupage «int(b) » est égal à e, le nombre recherché

Ligne 14 : introduit une fonction « IF » pour déterminer lorsque la longueur de la chaîne du résultat de 2**n est égale à la longueur de la séquence recherchée.

Ligne 15 : affiche par la fonction « print() » l'exposant « n ».

Ligne 16 : affiche par la fonction « print() » « b » (les derniers chiffres de la chaîne de caractères du résultat de 2**n).

Ligne 18 : implémente la fonction ELSE qui peut se traduire par 'sinon' pour contrebalancer avec la dernière fonction « IF » nommée.

Ligne 19 : création d'une nouvelle variable permettant de récupérer un caractère en plus de « a » par la longueur de la séquence cherchée.

Ligne 20 : affiche par la fonction « print() » l'exposant « n ».

Ligne 21 : affiche par la fonction « print() » la longueur de la séquence recherchée en ajoutant un caractère et les derniers chiffres « c ».

Ligne 22 : retour à la première fonction « IF » pour créer une variable nommée « g » qui est égale à 1.

Ligne 23 : utilisation de la variable « f » pour stocker le nombre de fois où le nombre entier du coupage a été égale à la longueur de la séquence recherchée (« int(b) = e ») a été vérifié.

Ligne 25 : sortie de la fonction « IF » et « FOR » pour implémenter une nouvelle fonction « IF » servant à vérifier que la variable « g » soit égale à 1.

Ligne 26 : affiche par la fonction « print() » la variable de stockage « f » (le nombre de fois où le nombre entier du coupage a été égale à la longueur de la séquence recherchée « e ») donc les ou l'exposant(s) trouvé(s) sur l'échantillon « r » puis le pourcentage des puissances se terminant par la séquence recherché « e ».

Ligne 27 : introduit une fonction « ELSE » pour contrebalancer avec la fonction « IF ».

Ligne 28 : affiche par la fonction « print() » ce qui ce trouve entre guillemets.

Quelques exemples du programme :

```
Séquence recherchée:16384
Exposant: 14
16384
Exposant: 2514
6 derniers chiffres: 616384
Exposant: 5014
6 derniers chiffres: 216384
Exposant: 7514
6 derniers chiffres: 816384
4 exposant(s) trouvé(s) sur l'échantillon de 10000 , 0.04 % des puissances
de termine par 16384
```

```
Séquence recherchée:97
rien de trouvé
```