DROLES DE TOUCHES

Année 2012

Rédacteurs : POTIER Fabien, Terminale S
BELLANGER Clément, Terminale S

Membres de l'atelier : BARRON Léa (TS), CASUSO Marion (TS), CESARO Jordan (1S), LOPEZ Anthony (Seconde) et MARCHAL Anthony (Seconde)

Lycée Saint Exupéry, 23 A Avenue de Metz, 57290 Fameck

Enseignants : BILLON Hélène et CANTUS Vincent

Chercheur : SCHOLLER Julie

I Introduction au problème :

A/ La calculatrice

Par erreur, notre calculatrice tombe et se détraque : elle affiche 0, et seules 4 touches fonctionnent encore. Les quatre touches sont les suivantes :

- Les touches 0 ; 4 ; 6 gardent leur fonction d'origine : le chiffre s'affiche à la droite du nombre affiché à l'écran.
- La touche 2 ne fonctionne plus de la même manière : quand elle est utilisée, si le nombre affiché est pair, elle le divise par 2 sinon elle ne fait rien.

Exemples :

À l'écran j'ai 14 et j'appuie sur la touche 0 : l'affichage devient 140.

À l'écran j'ai 12 et j'appuie sur la touche 2 : l'affichage devient 6.

À l'écran j'ai 13, j'appuie sur la touche 2 : 13

À l'écran, j'ai 9, j'appuie sur la touche 4 : 94

À l'écran, j'ai 7, j'appuie sur la touche 6 : 76

Lien vers la simulation de la calculatrice : zoo-nature.forum-actif.net/h8-la-calculatrice-cassee

Nous nous demandons s'il est encore possible d'afficher n'importe quel nombre sur la calculatrice.

B/ Introduction aux listes

DEFINITION :

Tout nombre entier n possède deux types de listes qui lui sont propres :

- **Les listes de touches :** ce sont les listes des touches sur lesquelles nous tapons pour afficher le nombre n.
- **Les listes d'affichages :** ce sont les listes des affichages intermédiaires que nous voyons sur la calculatrice avant d'arriver au nombre n.

Exemple :

Choisissons 16 et décomposons-le suivant ses listes :

- Une liste de touches de 16 est : (4 ; 2 ; 2 ; 6), la liste d'affichages correspondante est (4 ; 2 ; 1 ; 16).
- une autre liste de touches est (6 ; 4 ; 2 ; 2), la liste d'affichages qui correspond est (6 ; 64 ; 32 ; 16).

Remarques :

- Une liste (de touches ou d'affichages) est propre à un nombre, mais un nombre possède plusieurs listes de chaque type.
- Une liste d'affichage correspond à une liste de touches précise.

II Algorithme

A/ Présentation de l'algorithme

Un algorithme est une suite d'opérations à effectuer dans un ordre précis, comme une recette de cuisine. Tous les programmes informatiques proviennent d'un algorithme.

Notre algorithme nous donne la liste de touches et la liste d'affichages d'un nombre donné. Il fonctionne grâce aux réciproques des commandes de base de la calculatrice. Pour cela, nous avons étudié les fonctions de chaque touche de la calculatrice, elles sont les suivantes :

- Touche 0 : elle insère un 0 à la fin du nombre affiché, donc elle le multiplie par 10. Sa formule est 10x.
- Touche 2 : elle divise par 2 le nombre affiché s'il est pair. Sa formule est x/2.
- Touche 4 : elle insère un 4 à la fin du nombre affiché, donc elle le multiplie par 10 et lui ajoute 4. Sa formule est 10x+4.
- Touche 6 : elle insère un 6 à la fin du nombre affiché, donc elle le multiplie par 10 et lui ajoute 6. Sa formule est 10x+6.

Ensuite, il faut trouver les formules réciproques de celles des fonctions de base :

- Touche 0 : la réciproque de 10x est x/10 ou 0,1x
- Touche 2 : la réciproque de x/2 est 2x
- Touche 4 : la réciproque de 10x+4 est (x-4)/10 ou 0,1x-0,4
- Touche 6 : la réciproque de 10x+6 est (x-6)/10 ou 0,1x-0,6

Pour créer un procédé algorithmique, nous avons décidé pour raison de simplicité que lorsque le nombre se termine par 0, 4 ou 6, nous pouvons « faire tomber » ce chiffre (c'est-à-dire utiliser la formule réciproque associée à sa touche, et donc le faire disparaitre) et que lorsque l'un de ces chiffres n'est pas présent à la fin du nombre, nous le multiplions par 2 (réciproque de l'appui sur la touche 2). Donc on peut en déduire que le 2 est la touche qui revient le plus souvent.

L'algorithme est donc :

<u>Saisie :</u>
Affecter à x un nombre entier positif.

<u>Traitement :</u>
Tant que x est différent de 0 :
    Si x se termine par 0 :
        Noter 0.
        Faire tomber le 0.
    Sinon si x se termine par 4 :
        Noter 4.
        Faire tomber le 4.
    Sinon si x se termine par 6 :
        Noter 6.
        Faire tomber le 6.
    Sinon :
        Noter 2.
        Multiplier x par 2.
    Finsi
    Finsi
    Finsi
Noter à nouveau tous les chiffres dans l'ordre inverse.

<u>Sortie :</u>
Les chiffres obtenus représentent une liste de touches de x.

<u>Sur python :</u>

```
# -*-coding:Latin-1 -*
from math import *
from os import *
liste_H=[]
print("saisir un nombre")
nombrestr=input()
nombre=int(nombrestr)
```

```
while nombre!=0:
        if nombre/10==int(nombre/10):
                liste_H.insert(0,0)
                nombre=nombre/10
        elif (nombre-4)/10==((nombre-4)/10):
                liste_H.insert(0,4)
                nombre=(nombre-4)/10
        elif (nombre-6)/10==int((nombre-6)/10):
                liste_H.insert(0,6)
                nombre=(nombre-6)/10
        else:
                liste_H.insert(0,2)
                nombre=nombre*2
print (liste_H)
system("pause")
```

Exemple :

Nous allons trouver la liste de touches de 35 donnée par notre algorithme :

35
2x → touche 2
70
0.1x → touche 0
7
2x → touche 2
14
0.1x-0.4 → touche 4
1
2x → touche 2
2
2x → touche 2
4
0.1x-0.4 → touche 4
0

En lisant la suite de touches à partir du bas, on obtient la liste suivante : [4 ; 2 ; 2 ; 4 ; 2 ; 0 ; 2] donc on a bien obtenu une liste de touches de 35.

B/ Démonstration

La démonstration se fait par disjonction des cas. La démonstration par disjonction des cas consiste à séparer la totalité des nombres en plusieurs cas différents.

Nous séparons les nombres n en fonction de leur chiffre des unités. Nous commençons par compter le nombre de multiplications par 2 que nécessite chaque nombre pour se terminer par 0, 4 ou 6.

Exemple : prenons le cas d'un nombre n se terminant par 1.

Il ne se termine donc pas par 0, 4 ou 6, donc on le multiplie par 2 : on obtient alors un nombre se terminant par 2. Celui-ci ne se termine donc pas par 0, 4 ou 6 donc on le multiplie à nouveau par 2 : il se termine donc par 4, donc on peut « faire tomber » le 4.

Les nombres se terminant par 1 doivent donc être multipliés deux fois par 2 pour pouvoir enlever une unité. La formule à appliquer sur un nombre se terminant par 1 pour le faire baisser d'un rang est donc 0,1*((x*2)*2)-0,4 donc 0,4x-0,4.

On répète ensuite cette opération pour les 9 autres cas restants. On résume ensuite les résultats dans un tableau :

	0	1	2	3	**4**	5	6	7	8	9
x2		2	**4**	**6**		**0**	2	**4**	**6**	8
x2		**4**					**4**			**6**

Donc au bout de deux étapes au maximum, le nombre se terminera par 0, 4 ou 6. Ensuite, on cherche les formules à appliquer pour diminuer d'un rang, qu'on résume sous forme d'un tableau :

Chiffre	0	1	2	3	4
Formule	n/10 0,1n	(4n-4)/10 0,4n-0,4	(2n-4)/10 0,2n-0,4	(2n-6)/10 0,2n-0,6	(n-4)/10 0,1n-0,4
Chiffre	5	6	7	8	9
Formule	2n/10 0,2n	(n-6)/10 0,1n-0,6	(2n-4)/10 0,2n-0,5	(2n-6)/10 0,2n-0,6	(4n-6)/10 0,4n-0,6

On observe ensuite les formules ainsi découvertes : on remarque tout d'abord que ce sont des différences, donc elles sont inférieures à leur premier terme. Ensuite, ce terme lui-même est un produit entre le nombre n et un coefficient qui est toujours inférieur à 1 : donc ce terme est toujours inférieur à n et on peut en déduire que le résultat de la formule est toujours inférieur à n.

Finalement, si l'on utilise plusieurs fois ces formules de manière consécutive, avec un nombre entier naturel qui diminue strictement à chaque fois, on finit immanquablement par retomber sur 0. Ainsi l'algorithme se finit car on peut toujours trouver une liste finie de touches pour un nombre.

III A PARTIR D'UN AUTRE AFFICHAGE INITIAL

Après s'être interrogé sur les différentes listes à partir de 0, nous nous demandons s'il est possible de démarrer à partir d'un autre affichage initial.

Nous remarquons tout d'abord qu'une liste de touches avec l'affichage initial 0 commence obligatoirement par 4 ou 6 car lorsque nous avons 0, appuyer sur 2 ou sur 0 nous redonne le même nombre. Si, à partir d'un nombre donné, on arrive à afficher un 4 ou un 6, notre algorithme devient applicable.

DEFINITION :

Pour tout nombre n entier positif, on peut lui associer deux types de listes qui lui sont propres :

- Les listes de retour à 4 : c'est la liste des touches à taper pour réussir à afficher 4 à partir de n.
- Les listes de retour à 6 : c'est la liste des touches à taper pour réussir à afficher 6 à partir de n.

Exemple : l'exemple que nous avons le plus travaillé est celui dans lequel l'affichage initial est 13 :

- Une liste de retour à 4 de 13 est : (0 ; 4 ; 2 ; 2 ; 4 ; 2 ; 2 ; 2 ; 2 ; 2 ; 4 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2) mais aussi (6 ; 2 ; 2 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2 ; 4 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2 ; 4 ; 2 ; 2 ; 2 ; 2).
- Une liste de retour à 6 de 13 est : (6 ; 2 ; 2 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2 ; 4 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2)

Remarque :

Comme pour les listes de touches et d'affichages, une liste de retour est propre à un nombre précis, mais un nombre possède plusieurs listes de retour de chaque type.

Après quelques séances de recherches nous avons trouvé la liste de touche : (0 ; 4 ; 2 ; 2 ; 2 ; 6 ; 2 ; 2 ; 6 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2 ; 2) pour l'affichage initial 13. Nous avons alors cherché à savoir si on pouvait, à partir de n'importe quel nombre initial, revenir jusqu'à 4 (sachant que nous connaissons une liste de retour à 4 pour 6 : (1 ;2 ;2), l'étude de ce seul cas suffit). [1]

CONJECTURE : pour tout affichage initial n, il existe une liste de retour à 4.

Cela ne reste qu'une conjecture car nous n'avons pas trouvé de méthode générale pour trouver une telle liste. Cependant nous en avons trouvé une pour tous les entiers de 1 à 100.

IV D'AUTRES QUESTIONS QUE NOUS NOUS SOMMES POSEES

Au cours de nos recherches, nous avons aussi abordés les problématiques suivantes.

A/ Les bypass : le bypass du 6

Pour approfondir le sujet, la question s'est posée de savoir s'il était possible de faire fonctionner l'algorithme sans utiliser certaines touches de la calculatrice.

Tout d'abord, la question a été posée de savoir s'il était possible de retrouver des listes sans utiliser une seule fois la touche 6.

C'est possible car lorsqu'un nombre se termine par 6, après deux multiplications par 2, on en obtient un se finissant par 4. Pour les premiers tests, l'algorithme a été repris en enlevant juste la partie dans laquelle on fait tomber le 6 :

<u>Saisie :</u>
Affecter à x un nombre entier positif.
<u>Traitement :</u>
Tant que x est différent de 0 :
    Si x se termine par 0 :
        Noter 0.
        Faire tomber le 0.
    Sinon si x se termine par 4 :
        Noter 4.
        Faire tomber le 4.
    Sinon :
        Noter 2.
        Multiplier x par 2.
    Finsi
    Finsi
Noter tous les chiffres dans l'ordre inverse.

<u>Sortie :</u>
Les chiffres obtenus représentent une liste de touches de x.

On obtient ainsi un autre type de listes de touches, qui seront donc plus longues, mais tout aussi fonctionnelles.

<u>Démonstration :</u>

Il faut ensuite prouver que l'algorithme se termine, et ceci pour tous les nombres entiers positifs. Pour le faire, on utilise une démonstration similaire à celle du premier algorithme.

Tout d'abord, nous refaisons un tableau résumant les étapes de calcul en fonction du chiffre des unités :

	0	1	2	3	**4**	5	6	7	8	9
x2		2	**4**	6		**0**	2	**4**	6	8
x2		**4**		2			**4**		2	6
x2				**4**					**4**	2
x2										**4**

Ensuite, on recherche à nouveau les formules associées pour chaque chiffre des unités.

Chiffre	0	1	2	3	4
Formule	n/10 0,1n	(4n-4)/10 0,4n-0,4	(2n-4)/10 0,2n-0,4	(8n-4)/10 0,8n-0,4	(n-4)/10 0,1n-0,4
Chiffre	5	6	7	8	9
Formule	2n/10 0,2n	(4n-4)/10 0,4n-0,4	(2n-4)/10 0,2n-0,5	(8n-4)/10 0,8n-0,4	(16n-4)/10 1,6n-0,4

On observe ensuite les formules ainsi découvertes, on remarque tout d'abord que ce sont des différences. Pour les chiffres de 1 à 8, ces différences sont inférieures à n (voir démonstration dans le II).

On peut en déduire que l'affichage obtenu est inférieur à n. Il faut donc tester si, pour 9, la différence est inférieure à n. Pour cela, on utilise les congruences.

Un nombre a est congru à un nombre b modulo c si on peut écrire $a=b+k*c$ (avec k un nombre entier) on écrit : $a \equiv b$ (c).

Un nombre n qui se termine par 9 est congru à 9 ou à 19 modulo 20, en utilisant les propriétés sur les congruences, on peut écrire :

$n \equiv 9 \ (20)$ ou $n \equiv 19 \ (20)$

$16n \equiv 144 \ (20)$ ou $16n \equiv 304 \ (20)$

$16n \equiv 4 \ (20)$

$16n = 4+20k$ (avec k entier)

$16n-4=20k$

$(16n-4)/10=2k$

Ainsi, (16n-4)/10 est pair et c'est le nombre affiché après application d'une étape de la boucle de l'algorithme, sur un nombre se terminant par 9. Or un nombre pair se termine soit par 0, 2, 4 ou 6 ou 8. On peut donc étudier le résultat de deux applications de cette boucle sur un nombre qui se termine initialement par 9 :

on obtient à nouveau des fonctions affines de n. En étudiant les coefficients en n, ils sont tous inférieurs à 1 sauf pour un nombre se terminant par 8 consécutivement à la chute du 9.

Chiffres des affichages	9 puis 0	9 puis 2
Formule	0,16n-0,04	0,32n-0,48
Chiffres des affichages	9 puis 4	9 puis 6
Formule	0,16-0,44	0,64n-0,2

Cas problématique où les chiffres des affichages sont 9 puis 8 : 1,28n-0,72

On démontre, à l'aide de congruences, que le nombre (8n-4)/10 est pair si n est congru à 8 modulo 10, or un nombre pair se termine par 0, 2, 4, 6 ou 8. On peut donc étudier le résultat de trois applications de la boucle de l'algorithme sur un nombre qui se termine initialement par 9, puis par 8, en refaisant un autre tableau. De nouveau, un seul cas pose problème : si l'affichage se termine une fois encore par 8, le coefficient en n est 1,024.

On recommence en étudiant l'effet de quatre applications consécutives avec le même type de raisonnements que dans le paragraphe précédent et toutes les fonctions obtenues ont cette fois un coefficient inférieur à 1 (au maximum, c'est 0,8192)

CQFD

B/ La limite du 6

Après avoir utilisé les listes d'autant de manières différentes, nous nous sommes demandé si la longueur des listes fournies par notre algorithme pouvait être majorée. Après diverses recherches et différents tests, cette conjecture des listes a pu être trouvée :

CONJECTURE :

On définit un entier positif n, on note k le nombre de chiffres de ce nombre. La longueur de liste de touches fournie par notre algorithme (c'est-à-dire le nombre de touches à utiliser) ne dépasse pas 6k.

Exemple : On considère le nombre x=415, les chiffres qui le composent sont : 4-1-5, il est ainsi composé de 3 chiffres. Après application de l'algorithme, on trouve la liste : [4 ; 2 ; 2 ; 6 ; 6 ; 2 ; 0 ; 2]. Le nombre de touches utilisées est égal à 8 alors que $6\times3=18$. 8 étant bien plus petit que 18, le nombre d'éléments de sa liste de touches la plus courte est inférieure à la limite du 6.

Remarques :

- Cette propriété n'a pas pu être démontrée pour l'instant, mais elle a été vérifiée par informatique, tous les nombres jusqu'à $2*10^9$ sont compatibles avec celle-ci.
- Le premier nombre à atteindre cette limite est 87499.
- Les nombres possédant une limite proche de 6 se caractérisent par une récurrence de nombres se terminant par 1 ou par 9 dans leur liste d'affichage.

Notes d'édition

[1] La liste de retour à 4 pour 6 ne peut pas être (1, 2, 2) puisque la touche 1 ne fonctionne pas.
Une liste possible :(4, 2, 2, 2, 2)