Didactique de l'informatique

M1 MEEF NSI
Sur le rapport au nombre

Emmanuel Beffara

INSPÉ Institut national supérieur du professorat et de l'éducation
Académie de Grenoble

2022 – 2023

Plan

Conversion de base

Exercice

Écrire un programme permettant de convertir des nombres entiers de la base 10 à la base 2 et inversement.

Conversion de base

Exercice

Écrire un programme permettant de convertir des nombres entiers de la base 10 à la base 2 et inversement.

- Analyse a priori
- Exemples de productions
- Analyse a posteriori

Conversion de base

Une réponse proposée :

```python
def binaire(n):
    b = []
    while n != 0:
        b.insert(0, n % 2)
        n = int(n / 2)
    return b
```

Démonstration :

```
>>> binaire(17)
[1, 0, 0, 0, 1]
```

Conversion de base

Une réponse proposée :

```
def decimal(b):
    n = 0
    for i in range(len(b)):
        n += 2**i * b[len(b)-i-1]
    return n
```

Démonstration :

```
>>> decimal([1, 0, 0, 1, 1])
19
```

Conversion de base

Points de vigilance :

- le statut particulier de l'écriture décimale :
  Dans les deux réponses précédentes, la conversion depuis/vers la base dix est faite implicitement par l'interpréteur Python, pas par le programme.
- valeur renvoyée / valeur affichée
  cf. return ou print
- les implicites de la représentation

Plan

Une question d'arithmétique

Exercice

Soit n un entier. Dans la représentation en base 10 de $n!$, combien de zéros consécutifs se trouvent à droite ?

Une question d'arithmétique

Exercice

Soit n un entier. Dans la représentation en base 10 de $n!$, combien de zéros consécutifs se trouvent à droite ?

- À la main.
- (À la calculatrice.)
- Avec Python.

Plan

Un calcul de limite

Exercice

On pose la fonction

$$S_p(n) = \sum_{k=1}^{n} \frac{1}{p^k}$$

déterminer le comportement de $S_p(n)$ quand n tend vers l'infini.

Un calcul de limite

Exercice

On pose la fonction

$$S_p(n) = \sum_{k=1}^{n} \frac{1}{p^k}$$

déterminer le comportement de $S_p(n)$ quand n tend vers l'infini.

- À la main ?
- (À la calculatrice.)
- Avec Python.

Un calcul de limite

Exercice

On pose la fonction

$$S_p(n) = \sum_{k=1}^{n} \frac{1}{p^k}$$

déterminer le comportement de $S_p(n)$ quand n tend vers l'infini.

- À la main ?
- (À la calculatrice.)
- Avec Python.

Variables à considérer dans le calcul :

- La valeur de p : essayer 5, 6, 7, 8 ...

Un calcul de limite

Exercice

On pose la fonction

$$S_p(n) = \sum_{k=1}^{n} \frac{1}{p^k}$$

déterminer le comportement de $S_p(n)$ quand n tend vers l'infini.

- À la main ?
- (À la calculatrice.)
- Avec Python.

Variables à considérer dans le calcul :

- La valeur de p : essayer 5, 6, 7, 8 ...
- L'ordre dans lequel calculer la somme

Plan

Une suite récurrente

Exercice

On veut calculer u_{100} avec :

$$u_{n+1} = 4u_n - 1 \qquad u_0 = 1/3$$

Une suite récurrente

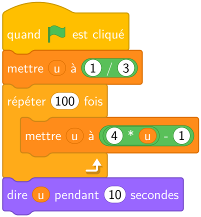
Exercice

On veut calculer u_{100} avec :

$$u_{n+1} = 4u_n - 1 \qquad u_0 = 1/3$$

```python
def recurrence(n):
    u = 1 / 3
    for i in range(n):
        u = 4 * u - 1
    return u

print(recurrence(100))
```



Une suite récurrente

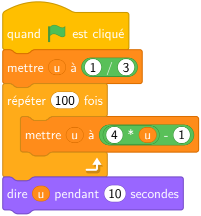
Exercice

On veut calculer u_{100} avec :

$$u_{n+1} = 4u_n - 1 \qquad u_0 = 1/3$$

```python
def recurrence(n):
    u = 1 / 3
    for i in range(n):
        u = 4 * u - 1
    return u

print(recurrence(100))
```



On trouve -2,9734326931374163e+43. **Pourquoi ?**

Une suite récurrente

0	0	24	0.328125	48	-1.4660155037e+12
1	0.333333333333	25	0.3125	49	-5.86406201480e+12
2	0.333333333333	26	0.25	50	-2.34562480592e+13
3	0.333333333333	27	0.0	51	-9.38249922369e+13
4	0.333333333333	28	-1.0	52	-3.75299968948e+14
5	0.333333333333	29	-5.0	53	-1.50119987579e+15
6	0.333333333333	30	-21.0	54	-6.00479950316e+15
7	0.333333333333	31	-85.0	55	-2.40191980126e+16
8	0.333333333332	32	-341.0	56	-9.60767920506e+16
9	0.333333333328	33	-1365.0	57	-3.84307168202e+17
10	0.333333333314	34	-5461.0	58	-1.53722867281e+18
11	0.333333333256	35	-21845.0	59	-6.14891469124e+18
12	0.333333333023	36	-87381.0	60	-2.45956587649e+19
13	0.333333332092	37	-349525.0	61	-9.83826350598e+19
14	0.333333328366	38	-1398101.0	62	-3.93530540239e+20
15	0.333333313465	39	-5592405.0	63	-1.57412216096e+21
16	0.33333325386	40	-22369621.0	64	-6.29648864383e+21
17	0.333333015442	41	-89478485.0	65	-2.51859545753e+22
18	0.333332061768	42	-357913941.0	66	-1.00743818301e+23
19	0.33332824707	43	-1431655765.0	67	-4.02975273205e+23
20	0.333312988281	44	-5726623061.0	68	-1.61190109282e+24
21	0.333251953125	45	-22906492245.0	69	-6.44760437128e+24
22	0.3330078125	46	-91625968981.0	...	...
23	0.33203125	47	-3.66503875925e+11	100	-2.97343269314e+43

Un calcul

Exercice

Faire effectuer le calcul suivant :

$$0,1 + 0,2$$

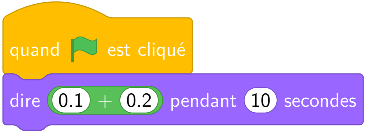
Un calcul

Exercice

Faire effectuer le calcul suivant :

$$0,1 + 0,2$$

```
>>> 0.1 + 0.2
0.30000000000000004
>>>
```



Plan

Les nombres à virgule flottante

La virgule flottante est une méthode d'écriture de nombres pas forcément entiers. Tous les nombres décimaux ne peuvent pas forcément s'écrire sous cette forme.

Codage en virgule flottante : $x = \pm m \times b^e$

- $\pm$: signe (1 bit),
- m : mantisse,
- b : base (en binaire, $b = 2$),
- e : exposant.

C'est comme la *notation scientifique*.

Virgule flottante – Standard IEEE 754

Simple précision : 32 bits

- 1 bit pour le signe,
- 8 bits pour l'exposant : $e - 127 \in [\![-126; +127]\!]$,
- 23 bits pour la mantisse.

Double précision : 64 bits

- 1 bit pour le signe,
- 11 bits pour l'exposant : $e - 1023 \in [\![-1022; +1023]\!]$,
- 52 bits pour la mantisse.

(Et quelques raffinements pour représenter 0, $\pm\infty$ et d'autres choses.)

Virgule flottante – Standard IEEE 754

Exemple 1

$$\underbrace{0}\ \underbrace{10000001}\ \underbrace{11100000000000000000000}$$

- Signe : 0, donc le nombre est positif.
- Exposant : $\overline{10000001}^2 = 129$ donc $e = 129 - 127 = 2$.
- Mantisse : $m = \overline{1,11100000000000000000000}^2 = 1,875$

Ainsi le code 01000000111100000000000000000000 représente le nombre décimal

$$+1,875 \times 2^2 = 7,5.$$

Virgule flottante – Standard IEEE 754

Exemple 2

$$\begin{aligned} 45,75 &= 2^5 + 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-2} \\ &= \overline{101101,11}^2 = \overline{1,0110111}^2 \times 2^5 \end{aligned}$$

- Signe : 0 car le nombre est positif,
- Exposant : $e = 5$, codé par $5 + 127 = 132 = \overline{10000100}^2$,
- Mantisse : $\overline{01101110000000000000000}^2$,

Ainsi $45,75$ est codé par

$$0\,10000100\,01101110000000000000000$$

Virgule flottante

Exercice

Voici un calcul fait par Python :

```
>>> 0.1 + 0.2
0.30000000000000004
>>>
```

En utilisant le codage en virgule flottante, expliquer le résultat donné.

Virgule flottante

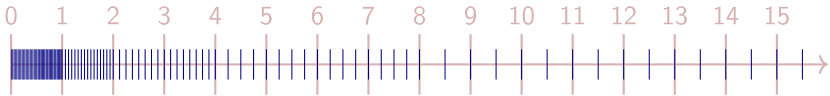
Exercice

Voici un calcul fait par Python :

```
>>> 0.1 + 0.2
0.30000000000000004
>>>
```

En utilisant le codage en virgule flottante, expliquer le résultat donné.

L'ensemble des flottants sur 8 bits (3 d'exposant, 4 de mantisse) :



Virgule flottante

Nous avons : $\overline{0,1}^{10} \approx \overline{0,000110110011001100110...}^{2}$

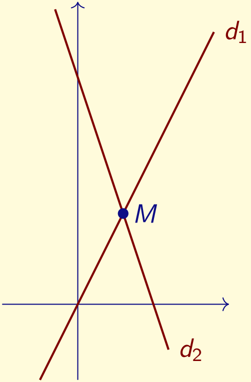
Le nombre décimal $0,1$ n'a pas d'écriture exacte en binaire en virgule flottante, comme de nombreux autres nombres.

Le calcul en virgule flottante impose de faire du calcul approché, donc le plus souvent ***l'égalité n'a pas de sens*** *pour ces nombres !*

Plan

Objets géométriques

Exercice



Calculer le point d'intersection des droites d_1 passant par $(0, 0)$ et $(1, 2)$ et d_2 passant par $(0, 3)$ et $(1, 0)$.

$$d_1 : y = 2x$$
$$d_2 : y = 3 - 3x$$

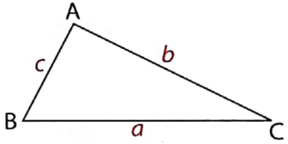
Vérifier le résultat avec Python et Geogebra.

Virgule flottante

Exercice

Voici l'exercice 15 p 335 du manuel *Hyperbole 2de - conforme aux aménagements de programme 2017* :

15 ABC est un triangle dont on connaît les longueurs, en cm, des côtés.
Écrire l'algorithme d'une fonction H(a, b, c) qui donne pour résultat la chaîne "Vrai" si le triangle ABC est rectangle en A et la chaîne "Faux" sinon.



Qu'en pensez-vous ?

Exercice

On donne l'algorithme suivant :

 # . . .
Saisir dans l'ordre croissant trois nombres entiers A, B, C
Affecter à X la valeur de $A^2 + B^2$
Affecter à Y la valeur de C^2
Si $X = Y$
    Alors
        Afficher . . .
    Sinon
        Afficher . . .
Fin Si

1. Compléter les trois lignes incomplètes de l'algorithme précédent.
2. Calculer les valeurs de X et Y pour $A = 7$, $B = 9$ et $C = 12$.
3. Quel est le résultat affiché à la sortie de l'algorithme dans ce cas ?
4. Écrire le script en Python correspondant à cet algorithme, et le tester.
5. Donner d'autres valeurs de A, B et C qui satisfont le test de sortie de l'algorithme.