

Feuille de TP Informatique 02

Calculs en bases ; Expressions, variables et types

Exercice 1.—

1. Donner l'écriture décimale de 10 1111. On pourra vérifier le résultat avec Python (en saisissant `0b101111` dans le shell).
2. Donner l'écriture en binaire naturel de 55. On pourra vérifier avec Python (saisir `bin(55)` dans le shell).
3. Donner l'écriture décimale de (écrit en base 8) 7413.
- 10 4. Donner l'écriture en base 3 de 63.

Exercice 2.—

1. Comment voir simplement si un entier écrit en binaire naturel est pair ou impair ?
2. Combien de chiffres 0 ou 1 sont nécessaires pour écrire en binaire naturel tous les chiffres d'une heure de la journée, au format heures-minutes-secondes, avec un nombre d'heures compris entre 0 et 23 ?
- 15 3. Quelle est la plus grande factorielle (nombres de la forme $n! = 1 \times 2 \times \dots \times (n-1) \times n$) que l'on peut écrire en binaire naturel avec 8 chiffres ? Avec 16 chiffres ?

Exercice 3.—

1. Donner l'expression en binaire naturel de 10 1100+1 1001 (sans passer par les décimaux).
2. Écrire la table en binaire naturel de tous les entiers compris entre 0 et 16 inclus (sans les calculer un par un, mais en ajoutant
- 20 1 à chaque fois).

Exercice 4.—

1. À quelle opération est équivalent le fait de décaler tous les bits d'un entier non signé vers la gauche, en rajoutant un 0 à droite (exemple : 10100 $\rightarrow$ 101000 ou encore 1001 $\rightarrow$ 10010) ?
2. À quelle opération est équivalent le fait de décaler tous les bits d'un entier non signé vers la droite, en supprimant le bit de
- 25 poids faible (exemple : 10100 $\rightarrow$ 1010 ou encore 1001 $\rightarrow$ 100) ?

Exercice 5.—

1. Quelle est l'écriture en décimal du nombre D2F en écriture hexadécimale ? On pourra vérifier le résultat avec Python (`0xD2F`).
2. Écrire sous forme hexadécimale le nombre 244. Vérifier avec Python (`hex(244)`).
3. Une clef Wifi s'écrit D336 45E4 4FA6 43BC 44F9 FA79 A6 en hexadécimal. Combien de chiffres 0 et 1 faudrait-il pour
- 30 l'écrire en binaire naturel ? Combien de chiffres nécessite l'écriture décimale de ce nombre (ne pas faire le calcul à la main : utiliser Python !) ? En déduire les avantages de la numération hexadécimale dans ce contexte.

Exercice 6.— Évaluer en Python les expressions suivantes –on pourra, au besoin, préalablement créer une ou plusieurs variables flottantes– en utilisant les fonctions mathématiques du module `numpy` permettant leur évaluation :

1. $\ln(10)$;
- 35 2. $e^{-1/(1-x^2)}$ avec $x = 0,5$;
3. $\binom{100}{50} = 100!/(50!)^2$ où $n! := n \times (n-1) \times \dots \times 2 \times 1$ désigne la factorielle de n (Vérifier que ce nombre est essentiellement le même que $\frac{2^{100}}{\sqrt{50 \cdot \pi}}$);
4. $k = 2\pi/\lambda$ avec $\lambda = 20$ cm.

Exercice 7.—

40 On considère l'extrait de script Python suivant, où les variables `a` et `b` sont de type numérique (des flottants par exemple) :

```
a = 4.125 # ou n'importe quel autre flottant
b = -3.88 # ou n'importe quel autre flottant
a = a+b
b = a-b
45 a = a-b
```

1. Par rapport à leurs valeurs initiales, quelles sont les valeurs de `a` et `b` après l'exécution du script ?
2. Vérifier la réponse à la question précédente en recopiant le script et en l'exécutant (rappel : dans IEP ou Spyder, l'exécution en bloc d'un script se fait à l'aide de la touche F5).
3. Pourquoi parle-t-on ici de calcul « sans mémoire additionnelle » ?
- 50 Comment aboutit-on au même résultat en utilisant un peu de mémoire additionnelle ?
4. En Python, comment peut-on, en une ligne, aboutir au même résultat ?

Exercice 8.— Donner le type et la valeur des expressions suivantes (de tête puis vérification en console) :

1. `44 >= 91//2`
2. `[(i**2+11)%7 for i in range(1,10)]`
3. `6400/3200`
- 5 4. `'Ce n\'est pas la peine'+ " de discuter là-dessus maintenant, Apollodore."`
5. `('Ce n\'est pas la peine'+ " de discuter là-dessus maintenant, Apollodore.") [3]`
6. `[8,9,10] [2]`
7. `int(5.6)`
8. `float(5)`
- 10 9. `len(str(455.8))`

Exercice 9.— Si `a`, `b` et `c` sont des variables entières, l'expression `a**b**c` est-elle égale à `(a**b)**c` ou à `a**(b**c)` ? Pour répondre, on pourra faire des tests dans le shell pour des valeurs arbitraires de `a`, `b` et `c`.

Qu'en est-il si l'on remplace `**` par `//` ?

Exercice 10.—

- 15 1. Prévoir les valeurs des expressions et vérifier sur l'ordinateur :

- 1.a. `False or False and not True`
- 1.b. `True or not True and False`
- 1.c. `17*19 != 324 or not True == False`
2. L'expression `1000 < 10 < 100` a-t-elle du sens en Python ? Même question pour `1000 > 10 < 100` ainsi que pour `1000 < 10 < 1/0`.
- 20 3. Selon Python, quelle est la plus grande des constantes `True` et `False` ? Expliquer comment vous avez obtenu votre réponse.
4. Soit `mot` une variable de type chaîne de caractères. Proposer une expression dont la valeur est `True` si `mot` contient toutes les voyelles de l'alphabet (`a`, `e`, `i`, `o`, `u`, `y`), et `False` sinon. Faire un test avec `mot` égal à

`"Portez ce vieux whisky au juge blond qui fume."`

et avec

`"A man, a plan, a canal : Panama."`

(qui au passage, sont respectivement un célèbre pangramme et un palindrome).

Exercice 11.—

1. On considère la chaîne de caractères `"Le neutron est aussi une onde."`, affectée à une variable dont l'identifiant est `chaine`.

- 25 Prévoir les valeurs des expressions suivantes, et vérifier ces prévisions à l'aide de Python :

- 1.a. `len(chaine)` 1.b. `chaine[5]` 1.c. `chaine[3:16]` 1.d. `chaine[3:16:4]`
- 1.e. `chaine[-1]` 1.f. `chaine[-2]` 1.g. `chaine[-3]` 1.h. `chaine[10:]`
- 1.i. `chaine[:10]` 1.j. `chaine[:]` 1.k. `chaine[::-1]`
2. Quelle est la valeur de l'expression `len("C\'est trop un truc de ouf.")` ?
- 30 3. Comment déclarer une chaîne contenant un `\` ?
4. Quelle est la différence entre les chaînes `"Je saute à la ligne"` et `"Je saute à la ligne\n"` ?

Exercice 12.—

1. On considère la liste `liste` égale à `[1,2,3,[4,5,6,7,8],[9,10,11,12],13,14,15,16]`. Prévoir les valeurs des expressions suivantes, et vérifier ces prévisions à l'aide de Python : 1.a. `len(liste)` 1.b. `liste[5]` 1.c. `liste[4]` [2]

- 35 2. Saisir et exécuter chacun des scripts suivants, expliquer ce qu'on observe et conclure sur une comparaison des deux scripts. (On reviendra plusieurs fois sur cette question subtile !)

2.a.

```
a = 10
b = a
40 a = 20
print(a,b)
```

2.b.

```
A = [10,20,30,40]
B = A
45 A[1] = 7
print(A,B)
```

Exercice 13.— On simule ici la génération d'une liste de bases azotées (guanine G, thymine T, adénosine A, cytosine C) sur un brin d'ADN.

1. Proposer une expression Python s'évaluant en une liste (`list`) constituée de 465 caractères égaux à 'G', 'T' et 'A' en alternance, *i.e.*

5 `['G', 'T', 'A', 'G', 'T', 'A', 'G', 'T', 'A', ...].`

2. Écrire un script ou une expression permettant d'engendrer la même liste que précédemment, mais en remplaçant chaque base par 'C' toutes les cinq bases, *i.e.*

`['C', 'T', 'A', 'G', 'T', 'C', 'G', 'T', 'A', 'G', 'C', 'A', ...].`

Exercice 14.— Le type `tuple`, très apparenté au type `list`, n'est pas explicitement au programme, mais on le rencontre souvent en pratique. Par exemple, la fonction `divmod` appliquée à deux entiers `a` et `b` retourne le couple (ou 2-uplet) constitué du quotient et du reste de la division euclidienne de `a` par `b` :

```
In [X]: T = divmod(17,3)
In [X+1]: print(T)
(5,2)
15 In [X+2]: type(T)
Out[X+2]: tuple
```

1. Dans Python, affecter à une variable `T` la valeur `(1,3.1415,False,"chaîne")`.

2. Quel est son type ?

3. Que valent `len(T)`, `T[0]` et `T[3]` ?

20 4. Tenter de remplacer l'élément d'indice 0 par -1. Que se passe-t-il ? Y a-t-il des points communs avec une liste ?

5. Comment créer un 1-uplet (n -uplet à un seul élément), par exemple le 1-uplet contenant 3.1415 comme seul élément ? L'expression `(3.1415)` suffit-elle ?

Exercice 15.— Python peut faire des calculs avec des nombres complexes, c.à.d. des nombres de l'ensemble noté $\mathbb{C}$ en mathématiques, avec les mêmes limites d'approximations qu'avec les flottants. On se propose ici de découvrir quelques possibilités de Python en termes de calculs sur les complexes.

25 1. Quel est le type de l'expression `1j` ?

2. Quelles sont les types et valeurs des expressions `1j**2` et `(1.+2j)*(-2.+9j)` ?

3. Soit `z` la variable `3+4j`. Que valent `abs(z)`, `z.real`, `z.imag` et `z.conjugate()` ? Que vaut l'argument principal du complexe `z` ? On pourra se renseigner (en ligne) sur le module `numpy`.