TP 1 : Entiers ; Premiers pas en Python

Exercice 1. Codage des entiers

1. Déterminer à la main l'entier représenté par $(10111)_2$ codé en naturel.

2. Vérifier à l'aide de python en utilisant la fonction ci-dessous :

`bin`(number)	renvoie un `str`	Renvoie l'expression binaire d'un entier précédé du préfixe '0b' (qui précise la base) Seuls les bits utiles sont affichés.

La fonction `int()` permet de définir un entier, même à partir de son expression binaire.

3. Déterminer en utilisant python l'entier représenté par $(101010101)_2$

Si un nombre est négatif, sa représentation binaire en complément à deux est indiquée par Python par un signe – devant le préfixe 0b. Exemple : -0b101010101

4. Déterminer en utilisant python la représentation binaire de -32.

Le professeur vous fournit deux fonctions : `cpltoint(s)` et `inttocpl(x,n)` qui convertissent respectivement un nombre binaire écrit sous forme de chaîne de caractères en complément à deux (sur le bon nombre de bits) et un nombre entier relatif sur n bits en complément à deux. Exécuter le script fourni.

5. Utiliser ces fonctions pour écrire -32 en complément à deux sur 8 bits.

Décoder $(1001\ 1101)_{CPL2}$

Exercice 2. Boucles inconditionnelles et commande range

1. Dans la console taper : `range(10)` et `list(range(10))`. Essayer également `list(range(3,22))` et `list(range(5,26,2))`

Comment générer la liste des nombres [1,6,11,16,21,...,96] ?

2. Essayer le script suivant, en respectant **l'indentation** :

```
for i in range(15):
    print(i)
```

Écrire un script qui permet d'afficher tous les nombres compris entre 3 et 100 pour lesquels le reste de la division euclidienne par 4 vaut 1

3. Essayer le script suivant, en respectant **l'indentation** :

```
a=0
for i in range(15):
    print(i)
    a=a+i
    print(a)
```

Écrire un programme qui calcule la quantité suivante et vérifier votre calcul à la main :

$$\sum_{k=0}^{k=100} (k^2 - 1)$$

Exercice 3. Boucles conditionnelles

Un autre type de boucles est couramment utilisé : il s'agit des boucles conditionnelles. Leur syntaxe générale est la suivante :

`while` *condition* :
    instructions

1. Réécrire le calcul de la question précédente à l'aide d'une boucle `while`
2. Trouver le plus petit entier n tel que :

$$\sum_{k=0}^{k=n} k^k > 10^7$$

3. On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par récurrence :

$$u_0 = 2 \text{ et } \forall n \in \mathbb{N},\ u_{n+1} = u_n^2 - 1$$

Écrire un programme qui permet de trouver la plus petite valeur de k pour laquelle $u_k > 10^6$

Exercice 4. Notion de fonction

Certains blocs d'instructions sont répétés plusieurs fois avec des paramètres différents. La notion de fonction, qui est plus large que celle utilisée en Mathématiques, permet d'éviter des réécritures inutiles de blocs de codes.

La syntaxe générale de fonction est :

`def` nom(paramètres) :
    instructions
    `return` valeur *# retourne une valeur et interrompt l'exécution de la fonction*

1. Essayer le script suivant et le commenter :

```
def f(x):
    print("essai 1")
    return x**2+1
    print("essai 2")

print(f(3))
```

2. Écrire une fonction `test` qui prend en argument un entier relatif n et renvoie le booléen `True` si n est positif ou nul et le booléen `False` sinon.
3. Écrire une fonction `somme` qui prend en argument deux entiers n et p et qui renvoie leur somme.
4. Écrire une fonction `longueur` qui prend en argument un entier strictement positif n et qui renvoie le plus petit entier k tel que $k^k > n$

Exercice 5. Instructions conditionnelles : if ... else

Il est possible de restreindre l'exécution de certaines instructions grâce à l'instruction conditionnelle `if`. Nous étudierons en détail les constructions possibles pour les instructions conditionnelles ultérieurement mais deux constructions sont pour le moment à connaître :

`if` *condition* :
    instructions

ou si l'on désire que des instructions soient réalisées alternativement si la condition n'est pas remplie :

`if` *condition* :
    instructions # exécutées si la condition est remplie
`else` : `instructions` *# exécutées sinon*

1. Écrire une fonction d'arguments deux nombres x et y qui renvoie le plus grand de ces deux nombres.
2. Écrire de deux manières différentes une fonction de trois arguments x,y et z qui renvoie le plus grand de ces trois nombres.
3. Écrire une fonction `valabs` d'argument x qui renvoie la valeur absolue de x.
4. Une année est bissextile si elle est divisible par 4, sauf si elle est divisible par 100 auquel cas elle ne l'est pas, sauf si elle est divisible par 400, auquel cas elle l'est. Écrire une fonction `bissex` d'argument n, qui représente un entier et qui renvoie un booléen indiquant si une année est bissextile.

Exercice 6. Exercice de synthèse

On désire trouver, à 10^{-2} près, le réel compris entre $-n$ et n, n étant entier, pour lequel la quantité $|x-a|+|x-b|+|x-c|$ est minimale, avec a,b et c réels.

À l'aide des fonctions précédentes, écrire une fonction `minimise`, d'arguments a,b,c et n qui renvoie ce réel.