

## REPRÉSENTATION DES NOMBRES ENTIERS - EXERCICES

### Exercice 1

On considère  $(5279)_{10}$ .

- Décomposez ce nombre avec une somme de puissances de 10 et attribuez à chaque rang le nom correspondant (9 est le chiffre des unités, 7....)
- Combien de nombres entiers naturels différents en base 10 peut-on créer avec les 4 chiffres 5, 2, 7 et 9 ? Justifiez. Donnez un exemple parmi ces entiers et décomposez-le à son tour comme au 1°.

### Exercice 2

« Il y a 10 sortes de gens au monde : ceux qui connaissent le binaire et les autres » Expliquez cette phrase.

### Exercice 3 : écrire en base 10

- Écrire en base 10 :  
a)  $(0101)_2$                       b)  $(110)_2$                       c)  $(11101001)_2$
- Écrire un programme en Python permettant de convertir en base 10 un nombre en base 2 écrit sur 8 bits. Tester avec les nombres précédents.
- Écrire en base 10 :  
a)  $(14C)_{16}$                                       b)  $(7FA)_{16}$   
c) A quel nombre en base 10 correspond le mot BEEF écrit en hexadécimal ?

### Exercice 4 : écrire en base 2

Dans chaque cas, convertir le nombre en base 2 en indiquant le nombre de bits nécessaires au codage.

- $(14)_{10}$                       2)  $(218)_{10}$                       3)  $(42)_{10}$                       4)  $(57)_{10}$

### Exercice 5 : écrire en base 16

Dans chaque cas, convertir le nombre en base 16

- $(14)_{10}$                       2)  $(218)_{10}$                       3)  $(42)_{10}$                       4)  $(57)_{10}$

### Exercice 6 : bases 2 et 16

- Vérifiez vos réponses des exercices 4 et 5 en convertissant les nombres en base 2 des questions 1) et 2) de l'exercice 4 en base 16 ; puis en convertissant les nombres en base 16 des questions 3) et 4) de l'exercice 6 en base 2 DIRECTEMENT.
- Convertir en base 16, sans passer par la base 10 :  
1)  $(1001010)_2$                       2)  $(100010001)_2$                       3)  $(10100100111110010)_2$
- Convertir en base 2, sans passer par la base 10 :  
1)  $(5A92E3)_{16}$                       2)  $(BAD)_{16}$                       3)  $(FACADE)_{16}$

### Exercice 7 : un peu de programmation...

- a) Créer un programme Python qui convertit un nombre en base 10 dans une base à demander à l'utilisateur. Votre programme devra être composé d'une fonction *conversion(n,b)* qui prendra en argument un nombre  $n$  (en base 10) et une base  $b$ , et qui rendra le nombre converti en base  $b$ . Vous appliquerez la suite de divisions euclidiennes vu en cours pour la conversion. Les « input » et « print » seront réservés au corps du programme.  
b) Tester ce programme avec différentes bases plus petites que 10, puis avec la base 16. Quel est le problème ?
- Pour les plus rapides*: créer un programme Python qui convertit en base 10 un nombre écrit en base 16.

### Exercice 8 : un peu d'arithmétique...

- Faire les calculs suivants et donner le résultat des opérations binaires sans passer par la base 10. A chaque fois, on précisera le nombre de bits nécessaires au codage du résultat en binaire.  
a)  $(1101)_2 + (111)_2$                       b)  $(1101)_2 \times (111)_2$                       c)  $(1111)_2 + (10)_2$
- Vérifier en repassant à la base 10 à l'aide du programme de la question 2 de l'exercice 7.
- a) *Pour les très très rapides* : créer un programme Python qui fait la somme de deux nombres donnés en base 2 sur 8 bits (sans passer par la base 10 bien sûr...) puis retourne le résultat en base 2 sur 8 (ou 9 bits si nécessaire)  
b) Vérifier votre programme avec les questions 1a) et 1c), puis vérifier également qu'il fonctionne avec la somme  $(11111111)_2 + (10010010)_2$