

Exercice 6 : bases 2 et 16

1. 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.
2. Convertir en base 16, **sans passer par la base 10** :
1) $(1001010)_2$ 2) $(100010001)_2$ 3) $(10100100111110010)_2$
3. Convertir en base 2, **sans passer par la base 10** :
1) $(5A92E3)_{16}$ 2) $(BAD)_{16}$ 3) $(FACADE)_{16}$

Exercice 7 : un peu d'arithmétique...

1. 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 à l'encodage du résultat en binaire.
a) $(1101)_2 + (111)_2$ b) $(1101)_2 \times (111)_2$ c) $(1111)_2 + (10)_2$
2. 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 $(11111111)_2 + (10010010)_2$

Exercice 8 : pour les fans de programmation

1. 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 ?
2. Créer un programme Python qui convertit en base 10 un nombre écrit en base 16.