

TP8 (objectif : Ecriture en binaire, en hexadécimal, opérations, codage d'un entier positif et d'un entier signé sur plusieurs octets, par exemple un octet)

$xxxx_2$ signifie que l'on a une écriture en binaire.

$xxxx_{16}$ signifie que l'on a une écriture en hexadécimal.

$xxxx_{10}$ signifie que l'on a une écriture en décimal (écriture usuelle des nombres).

Un chiffre binaire est soit 0 soit 1 et est appelé bit.

1 octet (ou byte en anglais) est constitué de 8 bits.

1 hexadécimal est codé avec 4 bits.

1) Ecrire 15_{10} et 4_{10} en binaire.

Effectuer l'addition, la multiplication et la division euclidienne à partir des écritures en binaire.

Vérifier que l'on obtient la même chose qu'avec l'écriture décimale.

2) Ecrire en décimal, le nombre $0,11_2$ écrit en binaire

Multiplier $0,11_2$ par 100_2 en binaire et vérifier que c'est la même chose qu'en décimal.

3) Avec 1 octet, combien d'entiers naturels peuvent être codés, et donner le plus petit et le plus grand.

4) Ecrire en hexadécimal $1011\ 0111_2 = 240_{10}$

Ecrire en binaire et en décimal $0E2A_{16}$

5) Avec l'invite de commande et la commande « ipconfig/all »

expliquer comment sont codées l'adresse mac (physique) et l'adresse IPv4(réseau) de la carte réseau Ethernet

6) Coder sur 1 octet les entiers signés 3 et -3 en complément à 2.

Combien d'entiers signés peut-on coder avec 1 octet.

Donner le plus petit et le plus grand.

Vérifier que 0 et -0 se codent de la même façon avec 1 octet et que $x+(-x)=0$ sur 1 octet où x est un entier signé.

7) On suppose qu'en mémoire un entier signé est codé sur 4 octets.

Combien d'entiers peuvent être ainsi codés et en donner le plus petit et le plus grand.

(Le nombre d'octets utilisés dépend du langage de programmation)

Aide avec python :

```
>>> bin(255)
'0b11111111'
>>> hex(255)
'0xff'
>>> int(0b11111111)
255
>>> int(0xff)
255
>>>
```