

## TP Informatique 14

 On rappelle qu'un script (fichier \*.py) doit être enregistré et exécuté (touche F5) pour que les fonctions saisies dans le script soient utilisables dans la console et que la combinaison de touches **Ctrl+C** permet de casser une boucle infinie.

### Exercice 1

On rappelle que tout entier  $n$  non nul admet une unique écriture binaire

$$n = \sum_{i=0}^{p-1} d_i 2^i \quad \text{avec} \quad (d_0, \dots, d_{p-1}) \in \{0, 1\}^{p-1} \times \{1\}$$

que l'on note

$$n = \langle d_{p-1}, d_{p-2}, \dots, d_0 \rangle$$

Le chiffre  $d_0$  est appelé *bit de poids faible* et le chiffre  $d_{p-1}$  est appelé *bit de poids fort*.

1. Écrire une fonction `binaire(n)` d'argument  $n$  un entier qui renvoie son écriture binaire sous forme de liste par puissance croissante de 2, le bit de poids faible à gauche et le bit de poids fort à droite. Si l'entier  $n$  est nul, la fonction renverra la liste vide. On pourra tester sur différentes valeurs d'entiers  $n$  que le résultat de `binaire(n)` est conforme à celui de `bin(n)` (instruction native de conversion en binaire).
2. Écrire une fonction `decimale(L)` d'argument  $L$  une liste contenant l'écriture binaire d'un entier qui renvoie sa valeur. On implémentera l'algorithme de Horner. On pourra tester sur différentes valeurs d'entiers  $n$  que le résultat de `decimale(binaire(n))` renvoie bien l'entier  $n$ .
3. Pour différentes valeurs de  $n$  entier, vérifier que la taille de l'écriture binaire de  $n$  est en  $\lfloor \log_2(n) \rfloor + 1$  et observer l'effet de la multiplication et du quotient par deux sur l'écriture binaire.

### Exercice 2

Le codage d'un entier  $n$  sur 8 bits est décrit par

$$n = \sum_{i=0}^7 d_i 2^i \quad \text{avec} \quad (d_0, \dots, d_7) \in \{0, 1\}^8$$

que l'on note

$$n = \langle d_7, d_6, \dots, d_0 \rangle$$

1. Écrire une fonction `dec_bin8(n)` d'argument  $n$  un entier et qui renvoie son écriture binaire codée sur 8 bits sous forme de liste par puissance croissante de deux. Si l'écriture binaire de  $n$  déborde de 8 bits, les bits additionnels sont perdus. Par exemple, l'appel `dec_bin8(3)` renvoie la liste `[1, 1, 0, 0, 0, 0, 0, 0]` et l'appel `dec_bin8(256)` renvoie la liste `[0, 0, 0, 0, 0, 0, 0, 0]` (débordement).
2. Écrire une fonction `bin8_dec(L)` d'argument  $L$  une liste contenant l'écriture binaire d'un entier codé sur 8 bits qui renvoie sa valeur. On implémentera l'algorithme d'Horner. On pourra tester sur différentes valeurs d'entiers  $n$  dans la plage `[0; 255]` que le résultat de `bin8_dec(dec_bin8(n))` renvoie bien l'entier  $n$ .

3. Écrire une fonction `add8(L1,L2)` d'arguments deux listes `L1`, `L2` contenant l'écriture binaire d'entiers codés sur 8 bits qui renvoie la liste de l'entier obtenue par addition des entiers codés par `L1` et `L2`. On n'utilisera pas l'opération d'addition +. En cas de débordement, le bit de poids fort est perdu.
4. Pour différentes valeurs d'entiers `a`, `b` dans la plage `[[0; 255]]`, afficher le résultat de `a+b` et le comparer à `bin8_dec(add8(ta,tb))` où `ta` et `tb` désignent les listes des écritures binaires sur 8 bits de `a` et `b`.

### Exercice 3

Le codage d'un *entier signé* (entier relatif) sur 8 bits en *complément à 2* est décrit par

$$n = -d_7 2^7 + \sum_{i=0}^6 d_i 2^i \quad \text{avec} \quad (d_0, \dots, d_7) \in \{0, 1\}^8$$

que l'on note

$$n = \langle d_7, d_6, \dots, d_0 \rangle_{\text{CPL2}}$$

1. Écrire une fonction `dec_sign8(n)` d'argument `n` un entier et qui renvoie son écriture binaire en méthode de complément à 2 codée sur 8 bits. Si l'écriture binaire de `n` déborde de 8 bits, les bits additionnels sont perdus. Par exemple, l'appel `dec_sign8(-1)` renvoie la liste `[1, 1, 1, 1, 1, 1, 1, 1]` et l'appel `dec_sign8(128)` renvoie la liste `[0, 0, 0, 0, 0, 0, 0, 0]` (débordement).
2. Écrire une fonction `sign8_dec(L)` d'argument `L` une liste contenant l'écriture binaire d'un entier signé codé en complément à 2 sur 8 bits qui renvoie sa valeur. On implémentera l'algorithme d'Horner. On pourra tester sur différentes valeurs d'entiers `n` dans la plage `[[ -128; 127 ]]` que le résultat de `sign8_dec(dec_sign8(n))` renvoie bien l'entier signé `n`.
3. Expérimenter la fonction `add8(L1,L2)` codée précédemment sur des listes `L1`, `L2` contenant l'écriture binaire d'entiers signés codés en complément à 2 sur 8 bits. On testera en particulier le calcul de `1 + (-1)`. Qu'observe-t-on ? Le codage en complément à 2 est-il judicieux ?