

# Interrogation d'ITC n° 4

Semaine du 09/06/2025

Durée : 35 minutes

Nom :

Prénom :

## Consignes

- Les codes doivent être présentés en mentionnant explicitement l'indentation, au moyen par exemple de barres verticales sur la gauche.
- Pour qu'un code soit compréhensible, il convient de choisir des noms de variables les plus explicites possibles.
- La performance du code proposé à chaque question entrera dans l'évaluation, de même que l'utilisation de boucles appropriées.
- Vous pouvez bien sûr utiliser une feuille de brouillon en parallèle.

20

## Exercice 1 Au stylo

1.  Donner, sur 8 bits, l'expression binaire des entiers (vus comme entiers positifs) : 1, 16, 153, 255. On justifiera ses réponses.

•  $1 = 1 \times 2^0$  donc  $1 = (00000001)_2$

•  $16 = 1 \times 2^4$  donc  $16 = (00010000)_2$

•  $153 = 76 \times 2 + 1$   
 $76 = 38 \times 2 + 0$   
 $38 = 19 \times 2 + 0$   
 $19 = 9 \times 2 + 1$   
 $9 = 4 \times 2 + 1$   
 $4 = 2 \times 2 + 0$   
 $2 = 1 \times 2 + 0$   
 $1 = 1 \times 2^0 + 0$

•  $255 = 2^8 - 1 = \sum_{k=0}^7 2^k = (1 \dots 1)_2$

donc  $153 = (10011001)_2$

2. On considère les listes ci-dessous :

$$L_1 = [0, 0, 0, 0, 0, 1, 0, 1], \quad L_2 = [1, 0, 1, 0, 1, 0, 1, 0].$$

- 2.1)  Donner la valeur de l'entier associé à  $L_1$  puis à  $L_2$ , vues comme des listes représentant des entiers positifs.

• Comme  $L_1[0] = 0$

$$m_1 = 2^0 + 2^2 = 5$$

$$m_2 = 2^1 + 2^3 + 2^5 + 2^7 = 170$$

- 2.2)  Donner la valeur de l'entier associé à  $L_1$  puis à  $L_2$ , vues comme des listes représentant des entiers relatifs.

• Comme  $L_1[0] = 0$ , la liste correspond à un positif, c'est  $m_1 = 5$

• Comme  $L_2[0] = 1$ , la liste correspond à un négatif, qui est  $170 - 2^8 = -86$

- 2.3)  Réaliser l'addition binaire des deux entiers précédents, et vérifier le résultat lorsque les listes représentent des entiers positifs.

$$\begin{array}{r} 00000101 \\ + 10101010 \\ \hline 10101111 \end{array} = (2^8 - 1) - 2^4 - 2^6 = 175$$

## Exercice 2 Conversions pour les entiers, programmation

### 1. [Codage des entiers naturels]

- 1.1) 2 Soient  $N$  un entier strictement positif et  $n \in [0; 2^N - 1]$ . Écrire une fonction **réursive**  $\text{nat2bin}(n:\text{int}, N:\text{int}) \rightarrow \text{list}$  qui renvoie la liste formant le codage binaire de  $n$  sur  $N$  bits.

```
def nat2bin (n: int, N: int) -> list
  if N == 0 :
    return []
  else:
    return nat2bin (n//2, N-1) + [n%2]
```

- 1.2) 2 Écrire une fonction **itérative**  $\text{bin2nat}(\text{Lbin}:\text{list}) \rightarrow \text{int}$  où  $\text{Lbin}$  est une liste d'éléments de  $\{0, 1\}$ , qui renvoie l'entier naturel dont  $\text{Lbin}$  est l'écriture binaire. On veillera à avoir une complexité linéaire en  $n = \text{len}(\text{Lbin})$ .

```
def bin2nat (Lbin: list) -> int :
  1 | n = 0
  2 | for e in L :
  3 |   n = 2*n + e
  4 | return n
```

- 1.3) 1 Justifier la complexité précédente, exprimée en fonction de  $n = \text{len}(\text{Lbin})$ .

$$C(n) = 1 + 3n = O(n)$$

l.1. ← boucle l 2/3

- 1.4) 3 Écrire une fonction  $\text{test}(L:\text{list}, p:\text{int}) \rightarrow \text{bool}$  avec  $L$  une liste représentant un entier naturel  $n$  codé sur  $N$  bits,  $p$  un entier avec  $1 \leq p < N$  et qui renvoie  $\text{True}$  si  $n < 2^p$   $\text{False}$  sinon. On commencera par expliquer en une phrase le test que vous souhaitez faire sur la liste, que l'on ne cherchera pas à démontrer mathématiquement.

\* Il faut que les chiffres devant  $2^{N-1}, 2^{N-2}, \dots, 2^p$  soient nuls, c'est à dire tester :  $\forall k \in [0; N-p-1], L[k] = 0$

\* def test (L: list, p: int) -> bool:

```
N = len(L)
  k = 0
  while k < N - p and L[k] == 0 :
    k += 1
  return (k == N - p)
```

### 2. [Codage des entiers relatifs]

- 2.1) 2 Soient  $N$  un entier strictement positif et  $n \in [-2^{N-1}; 2^{N-1} - 1]$ .

Écrire une fonction `relat2bin(n:int, N:int) -> list` qui renvoie la liste formant le codage binaire de l'entier relatif  $n$  sur  $N$  bits.  
On veillera ici à utiliser la technique complément à deux. On pourra se servir d'une fonction `addBin` codée en TP, et qui à deux listes binaires associe la liste binaire correspondant à la somme.



Voir TP Nombres

- 2.2) 

2
---

 Écrire la fonction `bin2relat(Lbin)` où `Lbin` est une liste d'éléments de  $\{0, 1\}$ , qui renvoie l'entier relatif dont `Lbin` est l'écriture binaire.  
Vous n'êtes pas obligé d'utiliser le complément à deux ici.



Voir TP nombres.