

# Binaire - TD

qkzk

## Binaire

### 1. Conversion binaire vers décimal.

Donnez les valeurs entières décimales représentées par les nombres :

- 0b101
- 0b10101
- 0b0101
- 0b00101
- 0b1101 1101
- 0b1001 0111
- 0b1011 1000

### 2. Examen d'une représentation binaire

On considère  $a = 0b1010\ 0110$  et  $b = 0b11\ 1101$

*Dans cet exercice on essaye d'éviter de passer par le décimal*

1. Lequel des deux est le plus grand ?
2. Ces nombres sont-ils divisibles par 2 ? Pourquoi ?
3. Combien de bits occupe la représentation binaire de  $a + b$  ?

### 3. Conversion décimal vers binaire.

1. Convertir les nombres suivants en binaire :

- 12
- 23
- 35
- 127
- 211
- 254
- 231

2. On considère des entiers représentés sur 1 octet. Quel est le plus grand entier représentable ?
3. Calculer mentalement les puissances de 2 jusque  $2^{20}$ .
4. Comparer les représentations binaires de 16 et 15 ; de 32 et 31 ; de 64 et 63 ?
5. Généralisons : quelles sont les représentations binaire de nombre de la forme  $2^k$  et  $2^k - 1$  ?
6. En remarquant que  $2\ 048 = 2^{11}$ , donner la représentation binaire de 2022.

```
2048 = 0b 1000 0000 xxxx
2047 = 0b  111 1111 xxxx
  25 = 0b          1 xxxx
2022 = 0b  111 1110 xxxx
```

#### 4. Binaire et python

Python permet d'obtenir la représentation binaire d'un entier à l'aide de la fonction `bin`. Voici ce qu'on obtient avec `help(bin)` :

```
Help on built-in function bin in module builtins:
```

```
bin(number, /)
    Return the binary representation of an integer.
```

```
>>> bin(2796202)
'0b10101010101010101010'
```

Inversement, la conversion d'une base  $b$  vers la représentation décimale s'obtient en passant à `int` une chaîne de caractères ainsi que la base.

Voici ce qu'on obtient avec `help(int)`

```
class int(object)
|   int([x]) -> integer
|   int(x, base=10) -> integer
|
|   Convert a number or string to an integer, or return 0 if no arguments
|   are given. If x is a number, return x.__int__(). For floating point
|   numbers, this truncates towards zero.
|
|   If x is not a number or if base is given, then x must be a string,
|   bytes, or bytearray instance representing an integer literal in the
|   given base. The literal can be preceded by '+' or '-' and be surrounded
|   by whitespace. The base defaults to 10. Valid bases are 0 and 2-36.
|   Base 0 means to interpret the base from the string as an integer literal.
|   >>> int('0b100', base=0)
|   4
```

1. Quelle instruction saisir pour obtenir la représentation décimale de `0b1101001` ?
2.  $x$  est un entier dont la représentation binaire est `110100`. Donner deux instructions différentes permettant d'obtenir sa représentation décimale.
3. Quel sera le résultat des instructions suivantes ?

```
>>> bin(123)
>>> int("0b1111")
>>> int("0b10101", 2)
>>> bin(0)
>>> int("0b101211", 2)
```

4. Python accepte la notation `0b110` pour représenter un entier, en l'occurrence 6...

Qu'obtient-on pour les opérations suivantes ?

```
>>> 0b101 + 2
>>> bin(0b110 + 0b1110)
```

#### 5. Capacité

Dépassement de capacité : si l'on représente des nombres sur une **taille fixe**, par exemple 8 bits, alors on ne peut dépasser `0b1111 1111 = 255`. Toute opération dont le résultat dépasse 255 provoque un *dépassement de capacité*.

1. Parmi les additions suivantes, lesquelles vont provoquer un **dépassement de capacité** lorsque les nombres sont encodés sur 8 bits ?

- 1111 1011 + 1001 1111
- 1001 1011 + 0111 1011
- 0011 1011 + 1001 1001
- 1010 1011 + 0001 0100

2. La taille d'une somme binaire nécessite de connaître les valeurs manipulées. Ce n'est pas le cas d'un produit.

Quelle sera le nombre de bits des valeurs suivantes ?

- $0110 \times 1100$
- $1111 \ 0011 \times 1101 \ 0101$