Opérateurs Booléens

- Par convention 0 est faux/false, 1 est vrai/true
- On peut retrouver facilement nand et nor à partir des tables de vérité and, or et not
- XOR renvoie 1 seulement si le nombre des entrées à 1 est impair

Exercice :
utiliser les tables de vérité pour prouver les égalités suivantes

1) $\overline{a+b} = \bar{a}.\bar{b}$
2) $\overline{a.b} = \bar{a}+\bar{b}$
3) $a \oplus b = (a+b).(\bar{a}+\bar{b})$
4) $a \oplus b = a\bar{b} + \bar{a}b$

non (not)

Notation algébrique : $\bar{a}$
Notation Python : not(...)
Table de vérité :

a	$\bar{a}$
0	1
1	0

and (et)

Notation algébrique : $a.b$
Notation Python : and
Table de vérité :

a	b	$a.b$
0	0	0
0	1	0
1	0	0
1	1	1

ou (or)

Notation algébrique : $a+b$
Notation Python : or
Table de vérité :

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	1

nand (non et)

Notation algébrique : $\overline{a.b}$
Notation Python : not (... and ...)
Table de vérité :

a	b	$\overline{a.b}$
0	0	1
0	1	1
1	0	1
1	1	0

nor (non ou)

Notation algébrique : $\overline{a+b}$
Notation Python : not(... or ...)
Table de vérité :

a	b	$\overline{a+b}$
0	0	1
0	1	0
1	0	0
1	1	0

xor (ou exclusif)

Notation algébrique : $a \oplus b$
Notation Python : ^
Table de vérité :

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Attention

$\overline{a+b} \neq \bar{a}+\bar{b}$

$\overline{a.b} \neq \bar{a}.\bar{b}$

⚠

Les lois de De Morgan nous disent au contraire que :

$\overline{a.b} = \bar{a}+\bar{b}$

$\overline{a+b} = \bar{a}.\bar{b}$