

*Exercice 1* : Teste dans la console Python  $(1+2)-3==0$  et  $(1.0+2.0)-3.0==0$ .

*Exercice 2* : Si  $n < 0$  vaut True et  $m < 0$  vaut True, que vaut  $n * m < 0$  ?

*Exercice 3* Détermine la valeur de *resultat* dans les cas suivants :

a)  $x = -2$

$resultat = x ** 2 == -4$

b)  $x = 3, y = 4, z = 5$

$resultat = x ** 2 + y ** 2 == z ** 2$

c)  $x = 3, y = -5$

$resultat = x ** 3 > 50$  and  $y ** 2 < 30$

d)  $x = 3, y = -5$

$resultat = x ** 3 > 25$  and  $y ** 2 < 30$

*Exercice 4* : Si  $a = 1$  et  $b = 0$ , que vaut l'expression  $\text{Not}(a \text{ Or } b) \text{ Or } (a \text{ And } b)$ .

*Exercice 5* : Donne la table de vérité de  $\text{Not}(\text{Not}(A) \text{ And } \text{Not}(B))$ . A quelle fonction booléennes correspond cette fonction ?

*Exercice 6* : Donne la table de vérité de  $\text{Not}(\text{Not}(A) \text{ Or } \text{Not}(B))$ . A quelle fonction booléennes correspond cette fonction ?

*Exercice 7* : Donne la table de vérité de  $(\text{Not}(A) \text{ And } B) \text{ Or } (A \text{ And } \text{Not}(B))$ . A quelle fonction booléennes correspond cette fonction ?

*Exercice 8* : Un octet comme 01011101 peu être considéré comme une succession de booléens FalseTrueFalseTrueTrueTrueFalseTrue. On veut pouvoir lire et/ou modifier chaque bit en comptant les bits par la droite bit 0, bit 1, ... , bit 7.

a) Ecris un programme transformant l'écriture en binaire d'un octet à celle en booléens.

b) Ecris une fonction changeant le bit 2 d'un octet.

c) Ecris une fonction changeant le bit de rang n.

d) Ecris une fonction changeant tous les bits d'un octet.

*Exercice 9* : La fonction de Sheffer exprime l'incompatibilité entre deux booléens. Elle est définie par la table ci-dessous :

$A$	$B$	$S(A; B)$
0	0	1
1	0	1
0	1	1
1	1	0

Montre que  $S(A; B) = \text{NOT}(A \text{ AND } B)$ .

Montre que  $\text{NOT}(A) = S(A; A)$ .

Montre que  $A \text{ OR } B = S(S(A; A); S(B; B))$ .

Que peut-on déduire des deux derniers résultats ?