Chapitre 6 : Opérations booléennes **spé ISN – TS**

I. Variable de type booléen

Une variable de type booléen est une variable qui ne peut prendre que deux valeurs : *vrai* (*true*) ou *faux* (*false*).
Le nom de ce type de variable vient du mathématicien britannique George Boole (1815-1864), créateur de la logique moderne, fondée sur une structure algébrique et sémantique, appelée aujourd'hui algèbre de Boole.

Pour coder les variables booléennes, on utilise par convention la valeur 0 pour *faux* et la valeur 1 pour *vrai*.
Nous noterons **B** l'ensemble {0;1} que nous appellerons ensemble des booléens.
Ce chapitre présentera quelques fonctions booléennes, *i.e.* Des fonctions d'une ou plusieurs variables dans **B** et à valeurs dans **B**.

On représentera les images données par chaque fonction grâce à une **table de valeurs**.

II. Comparaison de deux variables

Pour tous les types de variables (entiers, flottants, chaînes de caractère, booléens...), il est possible de comparer deux variables entre elles :

x == *y*	*x* est identique à *y*	(notation classique en maths : =)
x != *y*	*x* est différent de *y*	(notation classique en maths : ≠)

Le résultat de ce type d'expression est un booléen.

Exercice 1 : Donner le résultat (*vrai* ou *faux*) des comparaisons écrites en gras :

a. **2 == 9**
b. **4 != 2 × 2**
c. *x* = 2 ; *y* = 4 ; *z* = *x* × *y* × 3 ; **z != 18** ;
d. **(2 <= 4) == (3 > 5)**

Exercice 2 : Donner la valeur résultant des comparaisons écrites en gras :

a. **2 >= 5**
b. **(4 + 5) == (3 × 3)**
c. **length("abracadabra") == length("bricoleuses")**
d. **Typhanie == Typhaine**

Remarque : ne pas confondre l'utilisation du signe "=" pour une comparaison et pour l'affectation d'une valeur. En algorithmique, les deux commandes doivent être bien distinctes pour être interprétées correctement par le compilateur.

À savoir : L'affectation s'écrit "=" et la comparaison s'écrit "==".

III. Fonctions booléennes à une variable

Les fonctions NON, identité et constantes sont définie dans **B** et sont à valeurs dans **B**. Leurs valeurs sont données par la table suivante :

x	0	1
non(x) = $\bar{x}$	1	0
Id(x)	0	1
Z(x)	0	0
U(x)	1	1

La fonction NON transforme donc 0 en 1 et 1 en 0 ou encore *faux* en *vrai* et *vrai* en *faux*.
En C, la fonction NON est représentée par le symbole !. Par exemple : !(a < b) est équivalent à (a >= b). Classiquement en mathématiques, la notation est ¬x.

Exercice 3 : 0 et 1 étant des entiers, exprimer la fonction non(x) à l'aide des opérateurs usuels sur les entiers.

IV. Fonctions booléennes à deux variables

1. Fonction ET (AND)

La fonction ET est une fonction définie sur $\mathbf{B}^2$ et à valeurs dans **B**.
On peut résumer sa table de valeurs par l'équivalence : (et(x,y) vaut 1) si et seulement si (x et y valent tous les deux 1).
Autrement dit, (x et y) est vrai ssi x et y sont simultanément vrais.

x	0	1	0	1
y	0	0	1	1
et(x, y) = (x et y)	1	0	0	1

En C, la fonction ET est représentée par le symbole && (en maths : ^).
Par exemple : (a < b) && (c > d).

Exercice 4 : 0 et 1 étant des entiers, exprimer la fonction et(x, y) à l'aide des opérateurs usuels sur les entiers.

2. Fonction OU (OR)

La fonction OU est une fonction définie sur $\mathbf{B}^2$ et à valeurs dans **B**. On peut résumer sa table de valeurs par l'équivalence : (ou(x,y) vaut 1) si et seulement si (x vaut 1 ou y vaut 1).

Autrement dit, (x ou y) est vrai ssi au moins l'une des deux conditions est vérifiée.

x	0	1	0	1
y	0	0	1	1
ou(x, y) = (x ou y)	1	1	1	1

En C, la fonction OU est représentée par le symbole |. Par exemple : (a<b) | (c>d). En mathématiques : $\vee$.

Exercice 5 : 0 et 1 étant des entiers, exprimer la fonction ou(x, y) à l'aide des opérateurs usuels sur les entiers.

Exercice 6 : Exprimer les fonctions constantes Z(x) et U(x) à l'aide des fonctions et, ou et non.

3. Fonction OUX (XOR)

La fonction OUX (avec X pour exclusif) est une fonction définie sur $\mathbf{B}^2$ et à valeurs dans **B**. On peut résumer sa table de valeurs par l'équivalence : (oux(x,y) vaut 1) si et seulement si (seul x vaut 1 ou seul y vaut 1).

Autrement dit, (x oux y) est vrai ssi x est différent de y.

x	0	1	0	1
y	0	0	1	1
oux(x, y) = (x ou y)	1	1	1	0

En C, la fonction OUX est représentée par le symbole ^ (en maths : $\oplus$).

Par exemple : (a < b) ^ (c > d).

Exercice 7 : 0 et 1 étant des entiers, exprimer la fonction oux(x, y) à l'aide des opérateurs usuels sur les entiers.

Exercice 8 :

1. Exprimer la fonction définie sur **B** par x → oux(x, 0) à l'aide des fonctions booléennes de base.
2. Même question pour la fonction définie sur **B** par x → oux(x, 1).
3. En déduire une expression de la fonction oux à l'aide des fonctions booléennes de base.

À savoir : Pour tous x et y dans $\mathbf{B}^2$, x xor y = (x et non(y)) ou (non(x) et y).

V. Exercices de logique (à rendre)

Exercice 9 : Lois de De Morgan et réduction de la base

Auguste de Morgan (1806 – 1871) est un mathématicien et logicien britannique, fondateur avec G. Boole de la logique moderne.

1. À l'aide de tables de valeurs, démontrer les deux lois de De Morgan :
   a. Pour tous x et y appartenant à **B**, on a non(x) et non(y) = non(x ou y).
   b. Pour tous x et y appartenant à **B**, on a non(x) ou non(y) = non(x et y).
2. Déduire de la première loi que toute fonction booléenne peut s'exprimer uniquement à l'aide des fonctions non et et.
3. Déduire de la deuxième loi que toute fonction booléenne peut s'exprimer uniquement à l'aide des fonctions non et ou.

Exercice 10 : Fonction NAND de Sheffer

On définit la fonction de Sheffer sur $\mathbf{B}^2$ par S(x, y) = non(x et y) pour tous x et y dans **B**. Cette fonction est souvent appelée NAND.

1. Dresser la table de valeurs de cette fonction S(x, y).
2. Exprimer la fonction non(x) à l'aide de la fonction S.
3. Exprimer la fonction ou(x, y) à l'aide de la fonction S.
4. Déduire des questions précédentes et de l'exercice 9 que tout fonction booléenne peut s'écrire uniquement à l'aide de la fonction NAND.

À savoir : **toute fonction booléenne peut s'écrire uniquement à l'aide de la fonction NAND** (soit, en termes d'électronique, avec un seul type de portes logiques). Cette démonstration a été faite en 1913 par Henry Maurice Sheffer (1882-1964), logicien américain.

Exercice 11 :

Sur une planète vivent les Purs (qui disent toujours la vérité) et les Pires (qui mentent toujours). Vous croisez deux personnes, Ender et Valentine, sur cette planète. Valentine affirme « Au moins l'un de nous deux est un Pire. »

Notons V : « Valentine est pure. » et E : « Ender est pur. ».

À l'aide d'une table de valeurs, dire ce que sont Ender et Valentine.