

## Structure Conditionnelle (suite)

### Les opérateurs logiques

Il arrive souvent qu'une condition ne soit pas aussi simple que celle que nous venons de voir. Afin de pouvoir gérer ce type de condition, nous avons à notre disposition les opérateurs logiques suivant :

- ET
- OU
- NON

Nous pouvons voir comment les utiliser, grâce aux [tables de vérité](#) suivantes :

cond1	cond2	cond1 ET cond2
Faux	Faux	Faux
Faux	Vrai	Faux
Vrai	Faux	Faux
Vrai	Vrai	Vrai

cond1	cond2	cond1 OU cond2
Faux	Faux	Faux
Faux	Vrai	Vrai
Vrai	Faux	Vrai
Vrai	Vrai	Vrai

cond1	Non (cond1)
Faux	Vrai
Vrai	Faux

### Prenons un exemple !

Nous voulons gérer l'accès à une attraction, mais celle-ci est réservée aux personnes mesurant au moins 120 cms, mais au plus 200 cms. Le but de cet algorithme est donc de saisir la taille, et d'indiquer ensuite si l'accès est autorisé ou interdit !

En regardant les tables de vérité ci-dessus, il apparaît immédiatement qu'il nous faut utiliser l'opérateur **ET**. Ce qui nous donne l'algorithme suivant :

```
Variable taille : Entier
Début
    Afficher "Entrez la taille : "
    Saisir ( taille )
    Si ( taille >= 120 Et taille <= 200 ) Alors
        Afficher ( "Accès autorisé !" )
    Sinon
        Afficher ( "Accès refusé !" )
    FinSi
Fin
```

### Priorité

Il peut arriver que l'on ait à poser une condition impliquant plusieurs opérateurs, dans ce cas, l'évaluation du **Et** se fait avant le **Ou**. Mais pour sécuriser l'écriture d'une telle condition, vous pouvez utiliser des parenthèses.

### A vous de jouer !

Écrire un algorithme qui va permettre de saisir deux entiers **non nuls**, et d'indiquer immédiatement et **sans faire le calcul** si leur produit est positif ou négatif !



```
variable ent1 : Entier
Début
| Afficher ( "Entrez première valeur : " )
| Saisir ( ent1 )
| Afficher ( "Entrez deuxième valeur : " )
| Saisir ( ent2 )
| Si ( ( ent1 > 0 Et ent2 > 0 ) Ou ( ent1 < 0 Et ent2 < 0 ) ) Alors
| | Afficher ( "Positif" )
| Sinon
| | Afficher ( "Négatif" )
| FinSi
Fin
```

