Théorie – les conditions et

les opérations sur les variables

Par Patrick Lapointe

1er février 2016



Table des matières

[I. L’instruction SI … ALORS 3](#_Toc364845643)

[II. Qu’est-ce qu’une condition ? 6](#_Toc364845644)

[III. Nom des variables 9](#_Toc364845645)

[IV. Les opérations sur les variables 11](#_Toc364845646)

[V. Premier problème 19](#_Toc364845647)

[VI. Deuxième problème 22](#_Toc364845648)

[VII.En résumé… 25](#_Toc364845649)

[Annexe 1 – Corrigé de l’exercice sur les conditions 26](#_Toc364845650)

[Annexe 2 – Corrigé de l’exercice sur les variables 27](#_Toc364845651)

[Annexe 3 – Corrigé de l’exercice sur les opérations mathématiques 28](#_Toc364845652)

[Annexe 4 – Corrigé de l’exercice sur les booléens 30](#_Toc364845653)

Théorie sur les conditions et les opérations sur les variables

1. L’instruction SI … ALORS

Comme nous l’avons dans le module précédent, un algorithme permet d’exécuter une liste d’instructions les unes à la suite des autres.

Mais on peut aussi « dire » à un algorithme de n’exécuter des instructions que si une certaine condition est remplie. Cela se fait grâce à l’instruction SI...ALORS

La structure de cette instruction est la suivante :

…

**SI** « condition » **ALORS**

« Placez ici les instructions à exécuter si la condition est VRAI »

**FIN SI**

…

Par exemple :

Le message s’affiche uniquement lorsque la condition est vraie : c’est-à-dire quand la note est supérieure à 60.

**SI** Note > 60 **ALORS**

AFFICHER « Succès »

**FIN SI**

Il est aussi possible d’indiquer en plus à l’algorithme de traiter le cas où la condition n’est pas vérifiée. On obtient alors la structure suivante :

…

**SI** « condition » **ALORS**

« Placez ici les instructions à exécuter si la condition est VRAI »

**SINON**

« Placez ici les instructions à exécuter si la condition est FAUX »

**FIN SI**

…

Par exemple :

**SI** Solde < 500 **ALORS**

Le message s’affiche uniquement lorsque la condition est vraie. Dans le cas contraire, c’est le deuxième message qui s’affiche.

AFFICHER « Montant insuffisant»

**SINON**

AFFICHER « Opération effectuée »

**FIN SI**

Pour l’ordinogramme, voici les symboles à utiliser :

Si Prix > 500

Rabais = 0

Fin Si

Calculer Rabais = Prix \* 0,10

Remarquer que le côté **VRAI** est toujours à gauche. Le côté **FAUX** correspond au SINON dans un pseudocode.

**VRAI** **FAUX**

Comme il ‘y a pas d’instruction du côté **FAUX**, on rejoint directement le **FIN SI**.

Si Prix > 500

Fin Si

Calculer Rabais = Prix \* 0,10

**VRAI**  **FAUX**

1. Qu’est-ce qu’une condition ?

Pour l’instruction **SI … ALORS**, c’est une condition qui se trouve entre les deux mots et qui permette de poursuivre le traitement.

Il y a deux réponses possibles à une condition : soit **VRAI**, soit **FAUX**. Aucune autre réponse n’est possible!!! Voici des exemples de conditions :

**A > 26** Se dit la variable **A** est supérieur à 26

**Moyenne < 60** Se dit la variable **Moyenne** est inférieur à 60

**Total = Budget** Se dit la variable **Total** est égal à la variable **Budget**

**Montant >= Limite** Se dit la variable **Montant** est supérieure ou égale à la variable **Limite**

**Quantite <= Max** Se dit la variable **Quantité** est inférieure ou égale à la variable **Max**

**MontantDu <> Min** Se dit la variable **MontantDu** est différente de la variable **Min**

**Pratiquons maintenant…**

Complétez le tableau ci-dessous.

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Variable A** | **Variable B** | **Réponse** |
| Si A = B ALORS… | 20 | 20 | VRAI |
| Si A <> B ALORS… | 60 | 60 |  |
| Si A >= 70 ALORS… | 70 |  |  |
| Si 1 = 1 ALORS… |  |  |  |
| Si 50 =< B ALORS… |  | 50 |  |
| Si A = B ALORS… | ALLO | ALLO |  |
| Si A <> B ALORS… | TATA | tata |  |
| Si (A-50) > (B\*5) ALORS… | 100 | 10 |  |

Réponse à la fin du document…

Dans Algobox, une égalité s’écrit ==, par exemple : SI Debut == Fin ALORS…

**IMPORTANT** : Assurez-vous de comparer des pommes avec des pommes, des oranges avec des oranges. Pourquoi?

Quand vous élaborez une condition, comparez des **nombres** avec des **nombres**, des **caractères** avec des **caractères**, etc.

**Truc mnémotechnique**

Avez-vous des difficultés pour différencier **<** et **>** ?

Essayons de le comprendre avec les deux exemples suivants : 4 > 3 et 3 < 5

Imaginons que les symboles **<** et **>** sont des grandes bouches ouvertes. Ça signifie que la valeur numérique 4 qui est à gauche du signe est supérieur (ou plus grand que) à 3 qui lui est placé à droite du signe.

Le truc : **« le petit mange le plus grand ».**

En utilisant l’exemple ci-haut :

3 mange 4 donc 3 **est le plus petit** que 4 ;

3 mange 5 donc 3 **est plus petit** que 5.

.

1. Nom des variables

Afin d’assurer une bonne compréhension de vos algorithmes et du bon fonctionnement de vos programme, il y a quelques règles à respecter en ce a trait à l’appellation de vos variables :

* Utiliser un nom **unique** ;
* Utiliser un nom **représentatif** ;
* Choisir un identifiant relativement **court** ;
* **Ne pas utiliser** des caractères comme +, =, -, \*, (), &, %, @, #, etc.
* **Ne pas séparer** l’identifiant de la variable, utilisez le caractère **\_** **ou** alternez majuscules et minuscules ; et
* **Ne pas utiliser** d’accent (é, è, ç, etc.).

|  |  |
| --- | --- |
| **Bon exemple** | **Mauvais exemple** |
| TypeAchat | a |
| MaVariable | voicilenomdemavariable |
| PourcTaxe | %taxe |
| GrandTotal | GT |
| grd\_tot | grd tot |
| Depart | Départ |

**Pratiquons maintenant…**

Choisissez un identifiant de variable en respectant les règles ci-dessus :

|  |  |
| --- | --- |
| **Description de la variable** | **Réponse** |
| Effectuer la somme de tous les notes | SommeNotes |
| Calculer le montant total de toutes les taxes |  |
| Entrer le nom du légume |  |
| Afficher le montant de l’addition |  |
| Écrire dans un fichier le nombre de personnes composant la famille (adultes et les enfants) |  |
| Imprimer le pourcentage des taxes scolaires |  |
| Retenir l’heure de départ |  |
| Afficher la date du jour |  |

Des exemples de réponses à la fin du document…

1. Les opérations sur les variables
2. Affectation

Qu’ce qu’une affectation ? C’est en fait une opération des plus courantes. Il s’agit d’attribuer la valeur à une variable.

Exemple 1 : Note = 80

Exemple 2 : MonNom = "Patrick"

Exemple 3 : SuccesOuEchec = Somme / NbreExmen

Dans cet exemple, la valeur numérique et entière 80 est affectée à la variable **Note**. Observez que l’opérateur permettant d’affecter est **=**.

1. Opérations mathématiques

Les opérations possibles sur une variable varient selon le type de valeur qu’elle représente.

Petit rappel de vos cours de math. Parmi les opérations de base possibles sur une variable numérique, il y a bien sûr les opérateurs mathématiques. Ceci inclut **l’addition**, la **soustraction**, la **multiplication**, la **division**, **l’exposant** et le **modulo**.

Les opérations mathématiques en programmation respectent l’ordre de priorité des mathématiques traditionnelles, soit parenthèse, exposant, division et multiplication ex aequo, addition et soustraction ex aequo. Pour changer l’ordre de priorité, il faut utiliser les parenthèses. Le truc pour s’en souvenir est **PEDMAS** :

|  |  |  |
| --- | --- | --- |
| **Priorité** | **Symbole** | **Nom** |
| 1 | () | **P**arenthèse |
| 2 | ^ | **E**xposant |
| 3 | \* | **M**ultiplication |
| / | **D**ivision |
| 4 | + | **A**ddition |
| - | **S**oustraction |
| Spécial | Mod | **Modulo** |

Mais qu'est-ce qu'un modulo ?

Le **modulo** est une expression mathématique liée à la division.

Par exemple, **100/2 = 50**, c'est une division. **100 / 3 = 33,33333**... , c'est aussi une division, mais dans ce deuxième exemple, le résultat de la division n'est pas un nombre entier : il y a une **virgule** et des **chiffres après** **celle-ci**.

Il est possible de dire que 100/3 = 33, reste 1. Ou vais-je pêcher ce « 1 » ?

Simplement si j’additionne 33 + 33 + 33, j’ai 99. Il me reste qu’à ajouter 1 pour obtenir notre valeur de départ 100.

Le modulo est en quelque sorte le **reste de la division** et **c’est toujours une valeur entière** (pas de chiffre après la virgule).

**Pratiquons maintenant…**

Effectuez les opérations demandées en respectant les règles ci-dessus :

|  |  |
| --- | --- |
| **Opérations à faire** | **Réponse** |
| 1 + 2 \* 4 / 2 | 5 |
| (1 + 2) \* (4 / 2) |  |
| 2^3 |  |
| 8 MOD 3 |  |
| 2 + 8 MOD 3 \* 5 |  |

Les réponses sont à la fin du document…

1. Opération booléenne sur des valeurs

Nous avons vu précédemment comment on utilise l’instruction **SI … ALORS** et comment on construit une condition.

En programmation, les signes **<**, **>** et **=** sont aussi utilisés pour comparer des valeurs de même type entre eux, c’est ce que l’on nomme **une opération booléenne**.

Pourquoi le mot booléen ? Il signifie que le résultat ne peut être qu’une des deux valeurs suivantes : soit **vrai**, soit **faux**.

Toutefois, il existe d’autres opérateurs booléens.

* 1. La fonction logique ET

Le **ET logique** est une vérification de deux ou plusieurs conditions afin de retourner une valeur booléenne sur l’ensemble des critères. Il faudra que **l’ensemble des conditions soient vraies** de part et d’autre du **ET** pour que le résultat soit **vrai**.

**Exemple**

Entier = 20

**Faux**

Vrai

Faux

Condition 1

Condition 2

Nom = "Québec"

Vérification : SI Nom = "Ontario" ET Entier > 10 ALORS

Le retour de la vérification est **faux** parce que la vérification de la variable **Nom** est **fausse** et celle de de la variable **Entier** est **vraie** : en résumé, **faux ET** **vrai** a pour résultat **faux**.

Voici un tableau résumant les retours du **ET logique** :

|  |  |  |
| --- | --- | --- |
| Condition 1\Condition 2 | **Vrai** | **Faux** |
| **Vrai** | Retour vrai | Retour faux |
| **Faux** | Retour faux | Retour faux |

Dit autrement : vrai ET vrai = vrai

vrai ET faux = faux

faux ET vrai = faux

faux ET faux = faux

* 1. La fonction logique OU

Le **OU** **logique** est l’inverse du **ET logique**. Il retournera une valeur vraie **dès qu’une des conditions est vraie**.

Tableau du **OU logique** :

|  |  |  |
| --- | --- | --- |
| Condition 1\Condition 2 | **Vrai** | **Faux** |
| **Vrai** | Retour vrai | Retour vrai |
| **Faux** | Retour vrai | Retour faux |

Dit autrement : vrai OU vrai = vrai

vrai OU faux = vrai

faux OU vrai = vrai

faux OU faux = faux

Maintenant compliquons les choses. Le **ET** et le **OU logiques** peuvent être utilisés en combinaison avec plusieurs vérifications pour donner un seul retour.

Par exemple :

SI (3 = 3 **ET** 3 > 2) **ET** (12 <> 1 **OU** 3=2) **OU** (12 < 0 **OU** (3 = 5 ET 3 > 0 ET 0 = 0)) ALORS …

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (3 = 3 **ET** 3 > 2) | | | **ET** | (12 <> 1 OU 3=2) | | | **OU** | (12 < 0 **OU** (3 = 5 **ET** 3 > 0 **ET** 0 = 0)) | | | | | | |
| ( V | **ET** | V ) | **ET** | ( V | **OU** | F ) | **OU** | ( F | **OU** | (F | **ET** | V | **ET** | V)) |
| vrai | | | **ET** | vrai | | | **OU** | ( faux | **OU** | Faux ) | | | | |
| **OU** | faux | | | | | | |
| vrai | | | | | | | **OU** | faux | | | | | | |
| Retour VRAI | | | | | | | | | | | | | | |

Dans cet exemple, on observe à chaque ligne du tableau l’évolution des vérifications booléennes. On peut voir qu’à plusieurs endroits, la vérification a retourné une valeur fausse mais que les **ET** et **OU logiques** ont fini par retourner une valeur finale vraie.

**N’oubliez pas que la présence de parenthèses change les priorités.**

**Pratiquons maintenant…**

Voici quelques variables pour compléter l’exercice :

**A = 10**, **B = 20**, **C=30**, **Nom = « Bob »**

Complétez le tableau suivant :

|  |  |
| --- | --- |
| **Opérations à faire** | **Réponse** |
| SI A = 10 ET B > 20 ALORS …  10 = 10 ET 20 > 20  VRAI ET FAUX | FAUX |
| SI A = 10 ET ((B > 20) OU (C >= 30)) ALORS … |  |
| SI (A\*10) <= 100 ET (B/5 >= 4 ET (C MOD 9 = 2)) ALORS … |  |
| SI C / A >= 3 ET Nom = "bob" ALORS … |  |

Les réponses sont à la fin du document…

1. Premier problème
2. **Comprendre le programme**

Une école a besoin d’un programme permettant de gérer le prêt de livres aux élèves.

L’école ne peut pas prêter plus de 10 livres à un élève. Le programme devra lire le **nom** de l’élève et le **nombre** de livres qu’il doit emprunter.

**Indiquer** signifie la même chose qu’**Afficher**.

Si le nombre de livres dépasse 10, indiquez à la préposée qu’il y a un dépassement et afficher de combien de livre. Si le nombre de livre est inférieur à 2, affichez qu’il manque des livres à prêter.

1. **Compléter un tableau de valeurs/se donner un exemple**

Par exemple…

|  |  |  |
| --- | --- | --- |
| **Entrée (au clavier)** | | **En Sortie (afficher)** |
| **Nom** | **Nombre de livres** | **Message affichée** |
| AAA | 11 | Dépassement |
| BBB | 10 | <Aucun message> |
| CCC | 2 | <Aucun message> |
| DDD | 1 | Il manque des livres |

1. **Effectuer l’ordinogramme**

Début du programme

Déclarer Nom, NbLivre

Lire Nom, NbLivre

Si NbLivre > 10

Fin Si

Fin du programme

Afficher "Dépassement"

Si NbLivre < 2

Afficher"Il manque des livres"

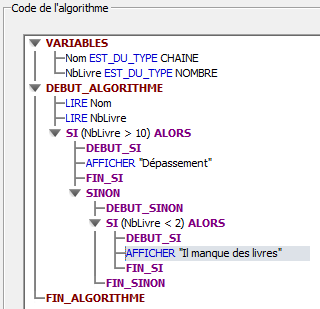
Fin Si

**VRAI** **FAUX**

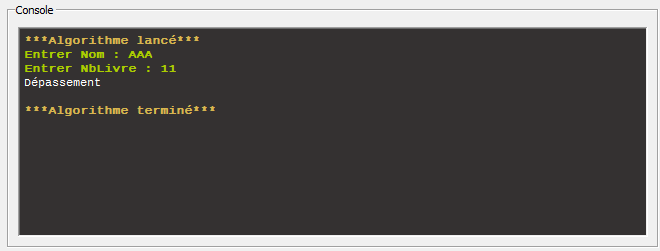
**VRAI**  **FAUX**

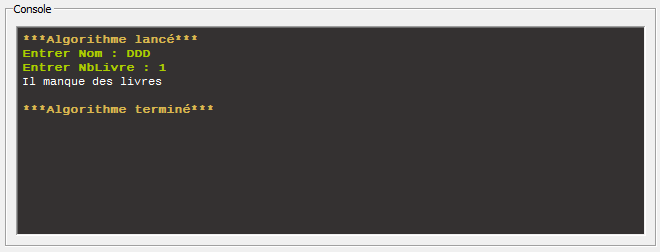
1. **Créer l’algorithme**

Utilise Algobox pour tester le programme.



1. **Tester l’algorithme**





1. Deuxième problème
2. **Comprendre le programme**

Une compagnie vous demande de concevoir un logiciel qui permet de traiter le chèque de retraite d’une personne.

Le **nom**, **l’âge**, la **province** et le **montant** du chèque sont entrés en mémoire.

Vous devez imprimer le chèque (un chèque a toujours un **nom** d’imprimer et un **montant**!) si la personne a 65 ans et plus **et** qu’elle demeure au Québec.

1. **Compléter un tableau de valeurs/se donner un exemple**

Par exemple…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entrée (au clavier)** | | | | **En Sortie (afficher)** |
| **Nom** | **Age** | **Province** | **Montant** | **Message affichée** |
| AAA | 11 | Ontario | 125$ | <Pas de chèque> |
| BBB | 70 | Alberta | 130$ | <Pas de chèque> |
| CCC | 64 | Québec | 135$ | <Pas de chèque> |
| DDD | 65 | Québec | 140$ | DDD 140$ |

1. **Effectuer l’ordinogramme**

Début du programme

Déclarer Nom, Age, Prov, Mt

Lire Nom, Age, Prov, Mt

Si Age >= 65 ET Prov = "Québec"

Fin Si

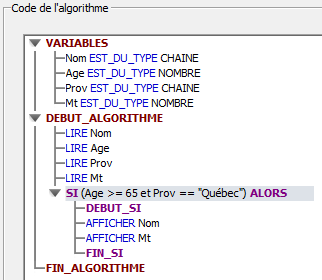
Fin du programme

Imprimer Nom, Mt

VRAI FAUX

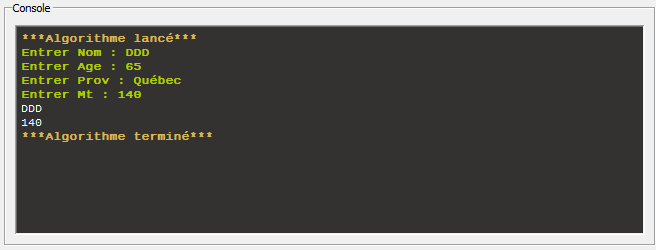
1. **Créer l’algorithme**

Utilise Algobox pour tester le programme.



1. **Tester l’algorithme**





1. En résumé…

* La valeur d’un booléen (une condition) est **VRAI** ou **FAUX**. (voir **II – Qu’est-ce qu’une condition ?**) ;
* Les **SI** peuvent être **imbriqués** : c’est-à-dire un **SI…ALORS** dans un **SI…ALORS**. (voir **premier problème**) ;
* **Il n’est pas obligatoire** d’avoir une instruction (comme AFFICHER, CALCULER, etc.) du côté **FAUX**/**SINON** d’un **SI…ALORS** ;
* Toutefois, **il est obligatoire** d’avoir une instruction du côté **VRAI** d’un **SI…ALORS** ;
* Dans un **pseudocode**, quand il n’y a pas d’instruction après le **SINON**, **il ne faut pas** écrire le **SINON**. (Voir **deuxième problème**) ;
* Dans un **ordinogramme**, quand il n’y a pas d’instruction du côté **FAUX**, **il faut relier** le **SI…ALORS** au **FIN SI**. (Voir **deuxième problème**).

Annexe 1 – Corrigé de l’exercice sur les conditions

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Variable A** | **Variable B** | **Réponse** |
| Si A = B ALORS…  20 = 20 | 20 | 20 | VRAI |
| Si A <> B ALORS…  60 <> 60 | 60 | 60 | FAUX |
| Si A >= 70 ALORS…  70 >= 70 | 70 |  | VRAI |
| Si 1 = 1 ALORS… |  |  | VRAI |
| Si 50 =< B ALORS…  50 =< 50 |  | 50 | VRAI |
| Si A = B ALORS…  « ALLO » = « ALLO » | ALLO | ALLO | VRAI |
| Si A <> B ALORS…  « «TATA » <> « tata » | TATA | tata | VRAI  En programmation, une lettre majuscule **n’est pas équivalente** à une lettre minuscule. |
| Si (A-50) > (B\*5) ALORS…  (100-50) > (10\*5)  50 > 50 | 100 | 10 | FAUX  N’oubliez pas la priorité des opérations : trouvez la réponse du calcul entre parenthèses avant. |

Annexe 2 – Corrigé de l’exercice sur les variables

Voici des **exemples** de réponses. Des réponses différentes sont acceptées dans la mesure qu’elles respectent les règles énoncées précédemment.

|  |  |
| --- | --- |
| **Description de la variable** | **Réponse** |
| Effectuer la somme de tous les notes | SommeNotes |
| Calculer le montant total de toutes les taxes | TotalTaxe  tot\_taxe |
| Entrer le nom du légume | Legume  NmLgme |
| Afficher le montant de l’addition | Mont  Montant  MtAdd |
| Écrire dans un fichier le nombre de personnes composant la famille (adultes et les enfants) | Tot\_Fmlle  TotalFam  Famille |
| Imprimer le pourcentage des taxes scolaires | PourTxScol |
| Retenir l’heure de départ | HDepart  Hre\_dep |
| Afficher la date du jour | Aujourdhui  DJour  Dat\_jr |

Annexe 3 – Corrigé de l’exercice sur les opérations mathématiques

|  |  |
| --- | --- |
| **Opérations à faire** | **Réponse** |
| 1 + 2 \* 4 / 2  1 + 8 / 2  1 + 4  5 | 5 |
| (1 + 2) \* (4 / 2)  3 \* 2  6 | 6 |
| 2^3  2 exposant 3  (3 indique le nombre de fois que l’on multiplie par lui-même)  2 \* 2 \* 2  8 | 8 |
| 8 MOD 3  8 / 3 = 2 reste 2  La preuve : 3 + 3 + 2 = 8  **Ou**  8 / 3 = 2,67  0,67 \* 3  2 | 2 |
| 2 + 8 MOD 3 \* 5  2 + 2 \* 5  2 + 10  12 | 12 |

Annexe 4 – Corrigé de l’exercice sur les booléens

Voici les variables pour compléter l’exercice :

**A = 10, B = 20, C=30, Nom = « Bob »**

|  |  |
| --- | --- |
| **Opérations à faire** | **Réponse** |
| SI A = 10 ET B > 20 ALORS …  **10** = 10 ET **20** > 20  **VRAI** ET **FAUX**  **FAUX** | **FAUX** |
| SI A = 10 ET ((B > 20) OU (C >= 30)) ALORS …  **10** = 10 ET ((**20** > 20) OU (**30** >= 30))  **VRAI** ET (**FAUX** OU **VRAI**)  **VRAI** ET **VRAI**  **VRAI** | **VRAI** |
| SI (A\*10) <= 100 ET (B/5 >= 4 ET (C MOD 9 = 2)) ALORS …  (**10**\*10) <= 100 ET (**20**/5 >= 4 ET (**30** MOD 9 = 2))  **100** <= 100 ET (**4** >= 4 ET (**3** = 2))  **VRAI** ET (**VRAI** ET **FAUX**)  **VRAI** ET **FAUX**  **FAUX** | **FAUX** |
| SI C / A >= 3 ET Nom = "bob" ALORS …  **30** / **10** >= 3 ET "**Bob**" = "bob"  **3** >= 3 ET **FAUX**  **VRAI** ET **FAUX**  **FAUX** | **FAUX** |