

Introduction à L'Informatique

Professeur responsable : Hervé Martin

Intervenants :

Manuele Kirsch-Pinheiro - Groupe II 1

Luiz-Angelo Estefanel - Groupes II 2 et II 3



Plan du Cours

- Logique binaire
 - NON (NOT)
 - ET (AND)
 - OU (OR)
 - OU-EXCLUSIF (XOR)
- Introduction à l'algorithmique
- Variables
 - Notion de variables
 - Notion de typage
 - Opérateurs basiques
 - Variables composées



Opérations Logiques

- On a vu précédemment que le système binaire permet des opérations arithmétiques
- Les chiffres binaires peuvent aussi participer à des opérations logiques :
 - NON
 - ET
 - OU
 - XOU



Opération Logique NON



- S'applique sur **un seul** opérateur
- Il **inverse** les valeurs des chiffres
NON 0 = 1
NON 1 = 0
- L'opération NON s'applique chiffre à chiffre
- NON 1010 = 0101



Opération Logique ET



- Vrai si tous les opérandes sont vrais

ET	0	1
0	0	0
1	0	1

0 peut être aussi considéré comme FAUX et 1 comme VRAI

- Les opérations s'appliquent chiffre à chiffre
- 1010 ET 0110 = 0010



Exercices



Quel est le résultat de :

NON 01 0011 =

0 0110 ET 1 0111 =

NON (1100 ET 1010) =

(NON 1100) ET 1010 =

1100 ET (NON 1010) =

(NON 1100) ET (NON 1010) =

Solutions



Opération Logique OU

- Vrai si au moins un des opérandes est vrai

OU	0	1
0	0	1
1	1	1

- On procède chiffre à chiffre
- 1010 OU 0110 = 1110

Exercices

Quel est le résultat de :

- 1010 OU 0011
- NON (1010 OU 1011)
- (1010 OU 0111) ET (1101 OU 0110)

Solutions

Opération Logique XOU

- Vrai si seulement un des opérandes est vrai

XOU	0	1
0	0	1
1	1	0

- On procède aussi chiffre à chiffre
- 1010 XOU 0110 = 1100

Exercices

Quel est le résultat de :

- 1010 XOU 0011 =
- NON (1010 XOU 1011)
- (NON 1010) XOU 0011
- (1010 XOU 0111) ET (1101 XOU 0110)

Solutions



Plan du Cours

- Logique binaire
 - NON (NOT)
 - ET (AND)
 - OU (OR)
 - OU-EXCLUSIF (XOR)
- Introduction à l'algorithmique
- Variables
 - Notion de variables
 - Notion de typage
 - Opérateurs basiques
 - Variables composées



Algorithme

- Définition : "Suite d'action à effectuer, dans un ordre donné, pour parvenir à un résultat"
- L'algorithme décrit l'environnement et les actions (logiques ou arithmétiques)
 - C'est la "recette" d'un programme



Exemple

• Recette de raclette

- Ingrédients :

- 200gr de fromage à raclette par personne
- 200gr de pommes de terre par personne
- cornichons, petits oignons blancs vinaigrés
- assortiment de charcuterie (jambon, viande séchée, rosette)



Exemple (cont.)

• Préparation :

- Faire cuire les pommes de terre
- Découper le fromage en tranches
- Présenter dans des vaisselles indépendants les cornichons, les oignons, la charcuterie
- Disposer une tranche de fromage dans chaque coupelle.
- Attendre quelques minutes, pour que le fromage fonde, grésille et dore.
- Disposer la raclette sur les pommes de terre, accompagnée du jambon et des différentes garnitures.



Analyse du exemple

• Environnement

- fromage à raclette, pommes de terre

• Actions

- Faire cuire les pommes de terre
- Couper le fromage en lamelles
- Disposer le fromage sur les pommes de terre



Autre exemple

- Changer une ampoule
 - Prendre une échelle
 - Dévisser l'ancienne ampoule
 - Prendre une nouvelle ampoule
 - Visser la nouvelle ampoule
 - Tester la nouvelle ampoule
 - Si elle ne s'allume pas, retourner au début



Exercices

- Faites un algorithme qui décrit la préparation d'un sandwich
 - Décrivez l'environnement et les actions
- Décrivez une procédure pour ouvrir une porte
 - Est-elle verrouillée ?
 - Et si vous aviez plusieurs clés, sans savoir quelle est la bonne ?



Plan du Cours

- Logique binaire
 - NON (NOT)
 - ET (AND)
 - OU (OR)
 - OU-EXCLUSIF (XOR)
- Introduction à l'algorithmique
- Variables
 - Notion de variables
 - Notion de typage
 - Opérateurs basiques
 - Variables composées



Variables

- Une **variable** est une entité qui **contient une information** :
 - une variable possède un nom, on parle d'**identifiant**
 - une variable possède une **valeur**
 - une variable possède un **type** qui caractérise l'ensemble des valeurs que peut prendre la variable
- L'ensemble des variables sont stockées dans la mémoire de l'ordinateur



Variables

- On peut faire l'analogie avec une armoire d'archive qui contiendrait des tiroirs étiquetés :
 - l'armoire serait la **mémoire** de l'ordinateur
 - les tiroirs seraient les **variables** (l'étiquette correspondrait à l'identifiant)
 - le contenu d'un tiroir serait la **valeur** de la variable correspondante
 - la couleur du tiroir serait le **type** de la variable (bleu pour les factures, rouge pour les bons de commande, etc.)



Actions sur les Variables

- On ne peut faire que deux choses avec une variable :
 1. **Obtenir son contenu** - regarder le contenu du tiroir
 2. **Affecter** un (nouveau) contenu - mettre une (nouvelle) information dans le tiroir
 - Cela s'effectue en utilisant l'opérateur d'affectation représenté par le **symbole =**



Actions sur les Variables

- Par exemple l'expression $c = a + b$ se comprend de la façon suivante :
 - On prend la valeur contenue dans la variable a
 - On prend la valeur contenue dans la variable b
 - On additionne ces deux valeurs
 - On met ce résultat dans la variable c
- Si **c** avait auparavant une valeur, cette dernière est perdue !



Type de données

- Le type d'une variable caractérise :
 - l'ensemble des valeurs que peut prendre la variable
 - l'ensemble des actions que l'on peut effectuer sur une variable
- Lorsqu'une variable est déclaré, on lui associe un type en utilisant la syntaxe suivante (Visual Basic)
 - Dim Identifiant de la variable As Type de la variable
- Par exemple :
 - Dim age As Integer
 - Dim nom As String
 - Dim prix As Single



Quelques types utiles en *Visual Basic*

- **Integer** - pour représenter des entiers
- **Single** - pour représenter des réels
- **String** - pour représenter des caractères
- **Boolean** - pour représenter des valeurs logiques (TRUE et FALSE)



Exercices

- Quel type de données est plus adapté aux valeurs suivantes?

Abc

1.5789

TRUE

29750

-10



Opérateurs

- Un **opérateur** est un symbole d'opération qui permet d'agir sur des variables ou de faire des "calculs"
- Un **opérande** est une entité (variable ou expression) utilisée par un opérateur
- Une **expression** est une combinaison d'opérateur(s) et d'opérande(s), évaluée durant l'exécution de l'algorithme, qui possède une valeur et un type



Opérateurs

- La signification d'un opérateur peut changer en **fonction du type** des opérandes
- Par exemple l'opérateur "+" avec des entiers aura pour sens l'**addition**, mais avec des chaînes de caractères aura pour sens la **concaténation**
 - $2+3$ vaut 5
 - "bonjour" + " tout le monde" vaut "bonjour tout le monde"



Opérateurs sur entiers ou réels

- On retrouve tout naturellement l'addition(+), soustraction(-), division(/) et la multiplication (*)
- Avec en plus pour les entiers l'opérateur **mod**, qui permet de calculer le reste de une division entière, par exemple :
 - $11 \bmod 2$ vaut 1



Exercices

- Faire un algorithme pour calculer la valeur de la division entière de deux nombres entiers
- Faire un algorithme pour calculer la valeur moyenne de deux numéros
- Faire un algorithme pour calculer une réduction de 10% sur un prix donnée

[Solutions](#)



Autres Opérateurs

- L'opérateur d'**égalité**
 - Représenté par le caractère =
 - Une expression contenant cet opérateur est un booléen
- On a aussi l'opérateur d'**inégalité** < >
- Et pour les types possédant un ordre les opérateurs de **comparaison** <, <=, >=, >



Exercices

- Faire un algorithme pour dire le montant d'impôt à payer, selon le revenu mensuel :
- 0-900€ -> 0€ d'impôt
- 901-5000€ -> 50€ d'impôt
- Plus de 5000€ -> 100€ d'impôt

Solutions



Variables Composées

- Souvent on a besoin de garder **plusieurs valeurs du même type**
 - Ex: une liste avec les ages des étudiants
- Il est prohibitif créer une variable pour chaque valeur
 - a, b, c, d, e, ..., a510
- On peut utiliser des **variables composées**, qui gardent plusieurs valeurs sous le même identifiant de variable



Vecteur

- Un vecteur est une variable qui contient plusieurs espaces pour garder les valeurs
 - Dim a(1 to 8) as Integer
- | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| a | 1 | 8 | 0 | 2 | 2 | 0 | 0 | 5 |
- On peut s'adresser à une position spécifique (5, par exemple) avec la syntaxe **a(5)=2**



Exercices

- Faire l'algorithme pour obtenir la moyenne d'age des étudiants de ce cours. On utilise un vecteur pour garder les ages des étudiants.

[Solution](#)

Matrices

- On peut aussi créer des matrices
- Ex: `Dim b(1 to 3,1 to 4) as String`

	1	2	3	4
b 1	A	a	B	b
2	C	c	D	d
3	E	e	F	f

- On adresse une position par `b(2,3) = D`

Exercices

- Déclarer une matrice pour garder les notes partielles des étudiants (3 notes) plus la moyenne finale.
 - Faire un algorithme pour calculer la moyenne finale de chaque étudiant, et la moyenne générale

[Solutions](#)

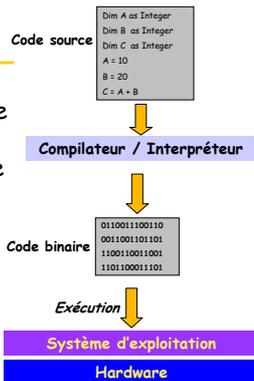
TPs Extras

- Ecrire un algorithme pour transformer les heures et les minutes en secondes
- Ecrire un algorithme pour transformer une température Celsius → Fahrenheit
 - $F = (C \div 5) \times 9 + 32$
- Ecrire un algorithme qui trouve la meilleure et la pire note d'un groupe d'étudiants
- Calcule la transposée d'une matrice



Visual Basic ?!?

- Visual Basic est un environnement et un langage de programmation
- On écrit les logiciels à l'aide des langages de programmations
 - Ex. : C, C#, Java, Perl...
- Comment ça marche ?



→ Retour

Organisation du Cours

✓ Solutions TPs



Solutions Exercices NON et ET



Quel est le résultat de :

$\text{NON } 01\ 0011 = 10\ 1100$

$0\ 0110\ \text{ET } 1\ 0111 = 0\ 0110$

$\text{NON } (1100\ \text{ET } 1010) = 0111$

$(\text{NON } 1100)\ \text{ET } 1010 = 0010$

$1100\ \text{ET } (\text{NON } 1010) = 0100$

$(\text{NON } 1100)\ \text{ET } (\text{NON } 1010) = 0001$

[Retour](#)



Solutions Exercices OU



Quel est le résultat de :

• $1010\ \text{OU } 0011 = 1011$

• $\text{NON } (1010\ \text{OU } 1011) = 0100$

• $(1010\ \text{OU } 0111)\ \text{ET } (1101\ \text{OU } 0110) = 1111$

[Retour](#)



Solutions Exercices XOU



Quel est le résultat de :

• $1010\ \text{XOU } 0011 = 1001$

• $\text{NON } (1010\ \text{XOU } 1011) = 1110$

• $(\text{NON } 1010)\ \text{XOU } 0011 = 0110$

• $(1010\ \text{XOU } 0111)\ \text{ET } (1101\ \text{XOU } 0110) = 1001$

[Retour](#)



Solution Exercices

- Faire un algorithme pour calculer la valeur de la division entière de deux nombres entiers

[Retour](#)

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
a = 15
b = 2
c = a \ b
```



Solution Exercices

- Faire un algorithme pour calculer la valeur moyenne de deux numéros

[Retour](#)

```
Dim a As Integer
Dim b As Integer
Dim moyen As Single
a = 15
b = 2
moyen = (a + b) / 2
```



Solution Exercices

- Faire un algorithme pour calculer une réduction de 10% sur un prix donnée

[Retour](#)

```
Dim prix As Single
prix = 10.0
prix = prix - (prix * 0.10)
```



Solutions Exercices

- Faire un algorithme pour dire le montant d'impôt à payer, selon le revenu mensuel :
- 0-900€ -> 0€ d'impôt
- 901-5000€ -> 50€ d'impôt
- Plus de 5000€ -> 100€ d'impôt



Solutions Exercices

Dim revenu As Single
Dim impot As Single

revenu = 1000

Si revenu <= 900 Alors
impot = 0

Sinon

Si revenu <= 5000 Alors
impot = 50

Sinon

impot = 100

Fin Si

Fin Si

[Retour](#)



Solutions Exercices

- Faire l'algorithme pour obtenir la moyenne d'age des étudiants de ce cours. On utilise un vecteur pour garder les ages des étudiants.

[Retour](#)

Dim ages (1 to 16) As Integer

Dim moyenne As Single

ages(1) = 20

ages(2) = 19

...

ages(16) = 21

moyenne = (ages(1) + ... + ages(16)) / 16



Solutions Exercices



- Déclarer une matrice pour garder les notes partielles des étudiants (3 notes) plus la moyenne finale.
 - Faire un algorithme pour calculer la moyenne finale de chaque étudiant, et la moyenne générale



Solutions Exercices



```
Dim notes (1 to 4, 1 to 16) As Integer
Dim moyenne As Single
notes(1,1) = 10
notes(1,2) = 14
...
notes(16,3) = 12
For i = 1 To 16 Step 1
    moyenne = (notes(i,1) + notes(i,2) + notes(i,3)) / 3
    notes(i,4) = moyenne
Next i
```

[Retour](#)