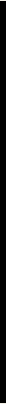
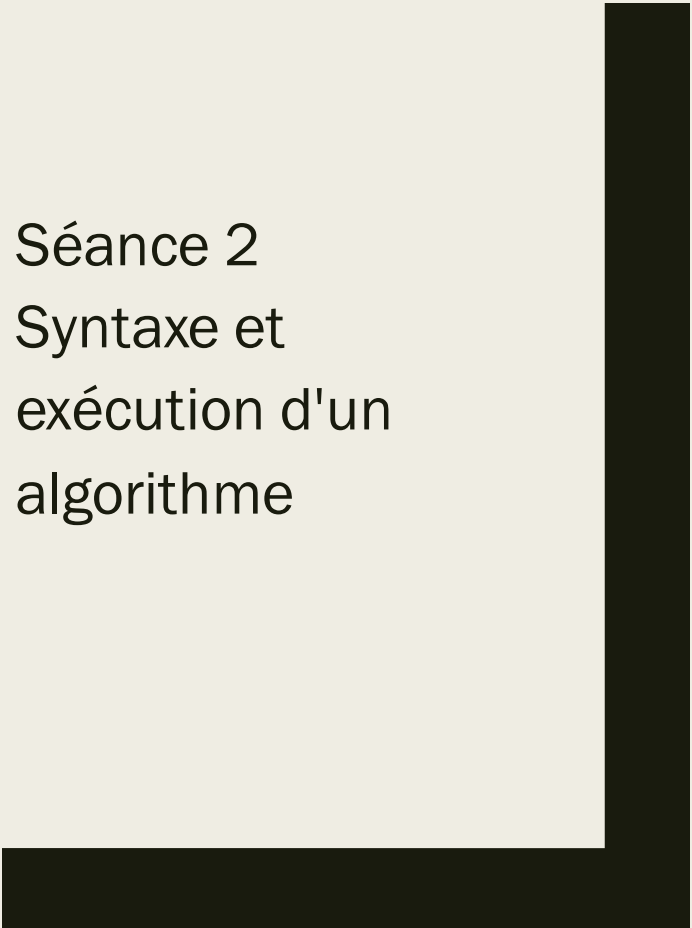




# ALGORITHMIQUE



Séance 2  
Syntaxe et  
exécution d'un  
algorithme



Camille SIMON – La Manu Le Havre

# Fonctionnement d'un algorithme

- "Un algorithme est un procédé définissant un ensemble de **suites d'instructions** à exécuter afin de **résoudre un problème** à partir des **données de départ** et d'arriver à un **résultat final** déterminé."
- Conception : création de l'algorithme avec spécification du contexte
- Exécution : utilisation de l'algorithme

### Algorithme *MoyenneNotes*

#### Variables

note : réel

moyenne : réel

#### Instructions

1. Ecrire("entrer la valeur de la note 1")
2. Lire(note)
3.  $\text{moyenne} \leftarrow \text{note}$
4. Ecrire("entrer la valeur de la note 2")
5. Lire(note)
6.  $\text{moyenne} \leftarrow \text{moyenne} + \text{note}$
7. Ecrire("entrer la valeur de la note 3")
8. Lire(note)
9.  $\text{moyenne} \leftarrow \text{moyenne} + \text{note}$
10.  $\text{Moyenne} \leftarrow \text{moyenne} / 3$
11. Ecrire("la moyenne est", moyenne)

# Syntaxe d'un algorithme

## ■ Rubriques :

- *Nom*

- *Variables*

- Déclaration des variables utilisées : nom et type

- Instructions

- Valeurs, variables et expressions

Algorithme *MoyenneNotes*

### Variables

**note : réel**

**moyenne : réel**

### Instructions

1. Ecrire("entrer la valeur de la note 1")
2. Lire(note)
3.  $\text{moyenne} \leftarrow \text{note}$
4. Ecrire("entrer la valeur de la note 2")
5. Lire(note)
6.  $\text{moyenne} \leftarrow \text{moyenne} + \text{note}$
7. Ecrire("entrer la valeur de la note 3")
8. Lire(note)
9.  $\text{moyenne} \leftarrow \text{moyenne} + \text{note}$
10.  $\text{Moyenne} \leftarrow \text{moyenne} / 3$
11. Ecrire("la moyenne est", moyenne)

# Syntaxe d'un algorithme

## ■ Rubriques :

- *Nom*

- *Variables*

- Déclaration des variables utilisées : nom et type

- Instructions

- Valeurs, variables et expressions

Algorithme *MoyenneNotes*

Variables

note : réel

moyenne : réel

Instructions

1. Ecrire("entrer la valeur de la note 1")
2. Lire(note)
3.  $\text{moyenne} \leftarrow \text{note}$
4. Ecrire("entrer la valeur de la note 2")
5. Lire(note)
6.  $\text{moyenne} \leftarrow \text{moyenne} + \text{note}$
7. Ecrire("entrer la valeur de la note 3")
8. Lire(note)
9.  $\text{moyenne} \leftarrow \text{moyenne} + \text{note}$
10.  $\text{Moyenne} \leftarrow \text{moyenne} / 3$
11. Ecrire("la moyenne est", moyenne)

# Syntaxe d'un algorithme

## ■ Rubriques :

- *Nom*

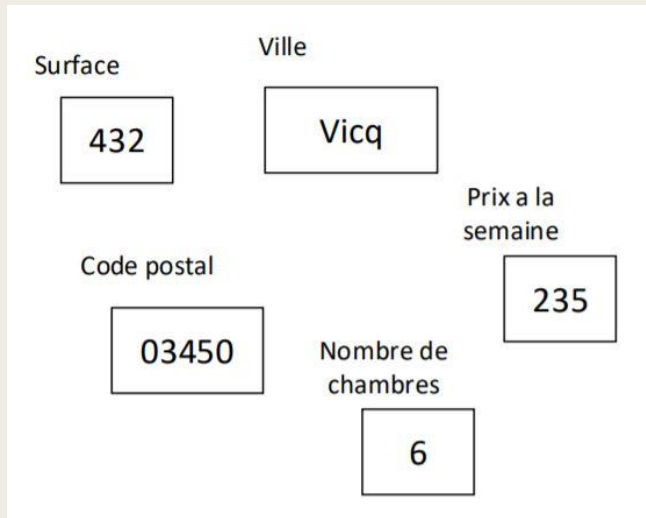
- *Variables*

- Déclaration des variables utilisées : nom et type

- Instructions

- Valeurs, variables et expressions

# Type de variables



- Chaque variable correspond à une case mémoire
- Stock une et une seule valeur
- Identifié avec un nom
- Types simples :
  - Entier : -1, 4, 9 852 354
  - Réel : 0,2, -4,31
  - Caractères : 'a', '^'
  - Booléen : vrai, faux

# Déclaration de variables

- Syntaxe :

`<nom> : <type>`

- `<nom>` doit être une suite de caractères


- Sans espace,
- Sans caractères spéciaux

- De préférence commençant par une minuscule

- Ne peut pas être un mot réservé (entier, variable, etc.)

- Exemples :

`somme, moyenneNotes, moyenne_notes`

 Dans ce cours, les noms sont en français. Dans le cadre professionnel, l'anglais est à privilégier.

# Opérations simples

- Opérations arithmétiques :

$+, -, *, /, E, \%$

- S'applique sur les réels et les entiers

- Division entière, noté E, est la partie entière de la division

$$E(7/2) = 3$$

$$E(24/5) = 4$$

- Modulo, symbolisé par %, est le reste de la division

$$7\%2 = 1$$

$$24\%5 = 4$$



# Opérations simples

- Opérations logiques:

$=, \neq, >, <, \geq, \leq$ , ET, OU, NON

- $=, \neq, >, <, \geq, \leq$  s'appliquent tous couples de variables de même type, retourne un booléen

$4 = 4$  donne *vrai*

$7 > 10$  donne *faux*

- ET, OU, NON s'appliquent sur des couples de booléen, retourne un booléen

Table ET		
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	faux

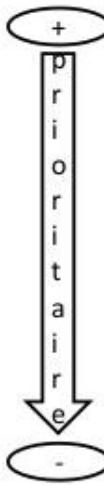
Table OU		
vrai	vrai	vrai
vrai	faux	vrai
faux	vrai	vrai
faux	faux	faux

Table XOR		
vrai	vrai	faux
vrai	faux	vrai
faux	vrai	vrai
faux	faux	faux

Table NON	
vrai	faux
faux	vrai

# Opérateurs

Associativité et priorités par ordre décroissant

A vertical arrow pointing downwards, labeled 'p r i o r i t a i r e' (precedence) on the left. At the top of the arrow is a circle containing a '+' sign, and at the bottom is a circle containing a '-' sign.

Catégorie	Opérateurs	Associativité
Référence	( ) [ ] . ->	GD
Unaire	+, -, NON	DG
Arithmétique	*, /, DIV, MOD	GD
Arithmétique	+, -	GD
Relationnel	<, ≤, >, ≥	GD
Relationnel	=, ≠	GD
Logique	ET	GD
Logique	OU	GD

29

# ORDRE DES OPÉRATIONS

# Fonctions

- Prend en entrée une ou plusieurs variables
- Retourne un unique résultat
- Résultat d'un ensemble d'opération
- Exemples :

Partie entière :  $E(<variable>)$

Division entière :  $E(<variable>/<variable>)$

Valeur absolue :  $Abs(<variable>)$

Minimum de deux valeurs :  $Min(<variable>, <variable>)$

Racine carrée :  $Sqrt(<variable>)$

# Expressions

- Combinaison d'éléments : valeurs, variables, opérations simples, fonctions.

- Exemple :

$3 * A * C + 7 * B * (C - 5) - \frac{1}{2} * D$

avec A, B, C, D : réel

$(X > Y) \text{ ET } (Y = Z) \text{ ET } (T > 'c')$

avec X, Y, Z : réel; T : caractère

$\text{abs}(x) + \cos(y) * \sin(z)$

avec x, y, z : réel

$z$

avec z : réel

- Règles :

- Concordance des types
- Règles usuelles de priorité de calcul

# Instructions - Affectation

- Affectation

$\text{<variable>} \leftarrow \text{<expression de même type que la variable>}$

- Evaluation de l'expression puis mémorisation du résultat dans la variable
- Si incompatibilité de type, il y a une erreur de syntaxe

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X-\frac{1}{2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?									
B	?									
C	?									
D	?									
X	?									
Y	?									
Z_positif	?									

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X-\frac{1}{2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1								
B	?	?								
C	?	?								
D	?	?								
X	?	?								
Y	?	?								
Z_positif	?	?								

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1							
B	?	?	1							
C	?	?	?							
D	?	?	?							
X	?	?	?							
Y	?	?	?							
Z_positif	?	?	?							



# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1						
B	?	?	1	1						
C	?	?	?	6						
D	?	?	?	?						
X	?	?	?	?						
Y	?	?	?	?						
Z_positif	?	?	?	?						

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1	-1					
B	?	?	1	1	1					
C	?	?	?	6	6					
D	?	?	?	?	'f'					
X	?	?	?	?	?					
Y	?	?	?	?	?					
Z_positif	?	?	?	?	?					

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1	-1	-1				
B	?	?	1	1	1	1				
C	?	?	?	6	6	6				
D	?	?	?	?	'f'	'f'				
X	?	?	?	?	?	2				
Y	?	?	?	?	?	?				
Z_positif	?	?	?	?	?	?				

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1	-1	-1	-1			
B	?	?	1	1	1	1	1			
C	?	?	?	6	6	6	6			
D	?	?	?	?	'f'	'f'	'f'			
X	?	?	?	?	?	2	2			
Y	?	?	?	?	?	?	-1			
Z_positif	?	?	?	?	?	?	?			

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow \text{'f'}$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < \text{'c'}$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1	-1	-1	-1	-1		
B	?	?	1	1	1	1	1	1		
C	?	?	?	6	6	6	6	6		
D	?	?	?	?	'f'	'f'	'f'	'f'		
X	?	?	?	?	?	2	2	2		
Y	?	?	?	?	?	?	-1	-1		
Z_positif	?	?	?	?	?	?	?	faux		

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow 'f'$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < 'c'$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1	-1	-1	-1	-1	-1	
B	?	?	1	1	1	1	1	1	1	
C	?	?	?	6	6	6	6	6	6	
D	?	?	?	?	'f'	'f'	'f'	'f'	'f'	
X	?	?	?	?	?	2	2	2	2	
Y	?	?	?	?	?	?	-1	-1	-1	
Z_positif	?	?	?	?	?	?	?	faux	faux	

# Instructions - Affectation

## Variables

X, Y, A, B, C : réels

D : caractère

Z\_positif : booléen

## Instructions

1.  $A \leftarrow -1$

2.  $B \leftarrow 1$

3.  $C \leftarrow 6$

4.  $D \leftarrow \text{'f'}$

5.  $X \leftarrow A+3$

6.  $Y \leftarrow 3*A*X+2*B*X*X^{-1/2}*C$

7.  $Z\_positif \leftarrow Y \geq 0$

8.  $Z\_positif \leftarrow D < \text{'c'}$

9.  $X \leftarrow X+1$

		1	2	3	4	5	6	7	8	9
A	?	-1	-1	-1	-1	-1	-1	-1	-1	-1
B	?	?	1	1	1	1	1	1	1	1
C	?	?	?	6	6	6	6	6	6	6
D	?	?	?	?	'f'	'f'	'f'	'f'	'f'	'f'
X	?	?	?	?	?	2	2	2	2	3
Y	?	?	?	?	?	?	-1	-1	-1	-1
Z_positif	?	?	?	?	?	?	?	faux	faux	faux

# Exercices - Affectation

## Exercice 1

Variables

A, B : entiers

Instructions

$A \leftarrow 1$

$B \leftarrow A+3$

$A \leftarrow 3$

## Exercice 2

Variables

A, B, C : entiers

Instructions

$A \leftarrow 5$

$B \leftarrow 3$

$C \leftarrow A+B$

$A \leftarrow 2$

$C \leftarrow B-A$

## Exercice 3

Variables

A, B : entiers

Instructions

$A \leftarrow 5$

$B \leftarrow A+4$

$A \leftarrow A+1$

$B \leftarrow A-4$

## Exercice 4

Variables

A, B, C : entiers

Instructions

$A \leftarrow 3$

$B \leftarrow 10$

$C \leftarrow A+B$

$B \leftarrow A+B$

$A \leftarrow C$

## Exercice 5

Variables

A, B : entiers

Instructions

$A \leftarrow 5$

$B \leftarrow 2$

$A \leftarrow B$

$B \leftarrow A$

Les deux dernières instructions permettent-elles d'échanger les valeurs de A et B ?

Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?

## Exercice 6

Ecrire un algorithme permettant d'échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu.

## Exercice 7

Une variante du précédent : on dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus de ces variables).



# Instructions - Ecriture

- Ecriture

Ecrire("<message à afficher>")

Ecrire("<message à afficher>", <variable>)

- Affiche à l'écran le message et la valeur de la variable

- Exemple :

Ecrire(" La note moyenne est : ", notmoyenne)

# Instructions - Lecture

- Lecture

Lire(<variable>)

- Inscrit la valeur entrée par l'utilisateur dans la variable
- Si la valeur entrée n'est pas du type de la variable, l'algorithme s'arrête : erreur d'exécution

# Exercices – Lecture et écriture

## Exercice 8

Variables

val, double : réels

Instructions

val  $\leftarrow$  231

double  $\leftarrow$  val\*2

Ecrire(val)

Ecrire(double)

## Exercice 9

Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.

## Exercice 10

Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

# Instructions - Conditions

- Alternative simple partielle

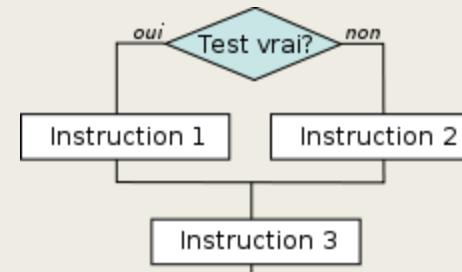
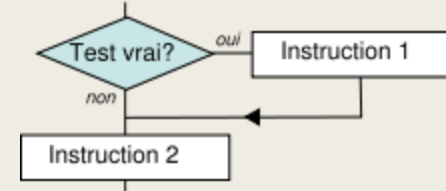
Si <expression booléenne> Alors  
    <instructions>  
FinSi

- Alternative simple complète

Si <expression booléenne> Alors  
    <instructions>  
Sinon  
    <instructions>  
FinSi

- Alternative multiple

Selon <expression booléenne> Alors  
    Cas <valeur 1> : <instructions>  
    Cas <valeur 2> : <instructions>  
    Cas <valeur 3> : <instructions>  
FinSelon



# Exercices - Conditions

## Exercice 11

Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

## Exercice 12

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit **pas** calculer le produit des deux nombres.

## Exercice 13

Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).

## Exercice 14

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !

## Exercice 15

Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

Peut-on concevoir plusieurs algorithmes équivalents menant à ce résultat ?

# Exercices - Logique

## Exercice 16

Cet algorithme est destiné à prédire l'avenir, et il doit être infaillible ! Il lira au clavier l'heure et les minutes, et il affichera l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre : "Dans une minute, il sera 21 heure(s) 33".

NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.

## Exercice 17

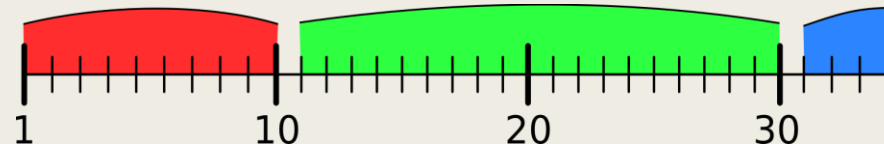
De même que le précédent, cet algorithme doit demander une heure et en afficher une autre. Mais cette fois, il doit gérer également les secondes, et afficher l'heure qu'il sera une seconde plus tard.

Par exemple, si l'utilisateur tape 21, puis 32, puis 8, l'algorithme doit répondre : "Dans une seconde, il sera 21 heure(s), 32 minute(s) et 9 seconde(s)".

NB : là encore, on suppose que l'utilisateur entre une date valide.

## Exercice 18

Un magasin de reprographie facture 0,10 € l'unité les dix premières photocopies, 0,09 € les vingt suivantes et 0,08 € au-delà. Ecrivez un algorithme qui demande à l'utilisateur le nombre de photocopies à effectuées et qui affiche la facture correspondante.



# Exercices - Logique

## Exercice 19

Les élections législatives, en Guignolerie Septentrionale, obéissent à la règle suivante :

- lorsque l'un des candidats obtient plus de 50% des suffrages, il est élu dès le premier tour.
- en cas de deuxième tour, peuvent participer uniquement les candidats ayant obtenu au moins 12,5% des voix au premier tour.

Vous devez écrire un algorithme qui permette la saisie des scores de quatre candidats au premier tour. Cet algorithme traitera ensuite le candidat numéro 1 (et **uniquement** lui) : il dira s'il est élu, battu, s'il se trouve en ballottage favorable (il participe au second tour en étant arrivé en tête à l'issue du premier tour) ou défavorable (il participe au second tour sans avoir été en tête au premier tour).

## Exercice 20

Ecrivez un algorithme qui après avoir demandé un numéro de jour, de mois et d'année à l'utilisateur, renvoie s'il s'agit ou non d'une date valide.

Cet exercice est certes d'un manque d'originalité affligeant, mais après tout, en algorithmique comme ailleurs, il faut connaître ses classiques ! Et quand on a fait cela une fois dans sa vie, on apprécie pleinement l'existence d'un type numérique « date » dans certains langages...).

Il n'est sans doute pas inutile de rappeler rapidement que le mois de février compte 28 jours, sauf si l'année est bissextile, auquel cas il en compte 29. L'année est bissextile si elle est divisible par quatre. Toutefois, les années divisibles par 100 ne sont pas bissextiles, mais les années divisibles par 400 le sont. Ouf !

# Instructions - Boucles

- Boucle déterminée avec un pas de 1

Pour <variable entière> de <expression entière> à <expression entière>

<instructions>

FinPour

- Exemple

Pour compteur de 1 à 10 faire

Ecrire(compteur)

FinPour



# Instructions - Boucles

- Boucle déterminée avec un pas différent de 1

Pour <variable entière> de <expression entière> à <expression entière> par pas de  
<expression entière>

<instructions>

FinPour

- Exemple

Pour compteur de 1 à 10 par pas de 2 faire

Ecrire(compteur)

FinPour

# Instructions - Boucles

- Boucle non déterminée

Tant que <expression booléenne> faire

    <instructions>

FinTantQue

- Exemple

compteur  $\leftarrow$  2

Tant que compteur%9  $\neq$  0 faire

    compteur  $\leftarrow$  compteur + 1

FinTQ

# Exercices - Boucles

## Exercice 21

Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.

## Exercice 22

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

## Exercice 23

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

## Exercice 24

Réécrire l'algorithme précédent, en utilisant cette fois l'instruction **Pour**.

## Exercice 25

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

Table de 7 :

$7 \times 1 = 7$

$7 \times 2 = 14$

...

$7 \times 10 = 70$

# Exercices - Boucles

## Exercice 26

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :

$$1+2+3+4+5=15$$

## Exercice 27

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée 8 !, vaut :

$$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$$

## Exercice 28

Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

etc.

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

C'était le nombre numéro 2

# Exercices - Boucles

## Exercice 29

Réécrire l'algorithme précédent, mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

## Exercice 30

Lire la suite des prix (en euros entiers et terminée par zéro) des achats d'un client. Calculer la somme qu'il doit, lire la somme qu'il paye, et simuler la remise de la monnaie en affichant les textes "10 Euros", "5 Euros" et "1 Euro" autant de fois qu'il y a de coupures de chaque sorte à rendre.

## Exercice 31

Écrire un algorithme qui permette de connaître ses chances de gagner au tiercé, quarté, quinté et autres impôts volontaires. On demande à l'utilisateur le nombre de chevaux partants, et le nombre de chevaux joués. Les deux messages affichés devront être :

Dans l'ordre : une chance sur X de gagner

Dans le désordre : une chance sur Y de gagner

X et Y nous sont donnés par la formule suivante, si n est le nombre de chevaux partants et p le nombre de chevaux joués (on rappelle que le signe ! signifie "factorielle", comme dans l'exercice 5.6 ci-dessus) :

$$X = n! / (n - p)!$$

$$Y = n! / (p! * (n - p)!)!$$

NB : cet algorithme peut être écrit d'une manière simple, mais relativement peu performante. Ses performances peuvent être singulièrement augmentées par une petite astuce. Vous commencerez par écrire la manière la plus simple, puis vous identifierez le problème, et écrirez une deuxième version permettant de le résoudre.