

ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS.
Mathématiques
Alg-C

Programmation langage C

Introduction à l'algorithmique

A.BELCAID

anasss.belcaid@gmail.com



Salle 1, TD 2 / 22 février 2016

- ▶ Thomas H. CORMEN et al. *Introduction to Algorithms*. 2nd. McGraw-Hill Higher Education, 2001. ISBN : 0070131511.
- ▶ Claude DELANNOY. *Le guide complet du langage C*. Broché, 2014.
- ▶ Claude DELANNOY. *Programmer en langage C : Cours et exercices corrigés*. Broché, 2014.
- ▶ Brian W. KERNIGHAN. *The C Programming Language*. Sous la dir. de Dennis M. RITCHIE. 2nd. Prentice Hall Professional Technical Reference, 1988. ISBN : 0131103709.

1. INFORMATIQUE

1.1 Définition

1.2 Codage

2. ALGORITHMIQUE

2.1 Définition

2.2 Caractéristiques

2.3 Exemples

2.4 Complexité

2.5 Types de bases

2.6 Variables

2.7 Déclaration

2.8 Affectation

2.9 Entrée/Sortie

2.10 Exercices

Informatique

Definition

Une **Information**

Definition

Une **Information**, est une **donnée** susceptible d'être **traitée**, transférée ou sauvegardée.

Definition

Une **Information**, est une **donnée** susceptible d'être **traitée**, transférée ou sauvegardée.

Definition

Un **Traitement**

Definition

Une **Information**, est une **donnée** susceptible d'être **traitée**, transférée ou sauvegardée.

Definition

Un **Traitement** est un ensemble d'opérations effectuées sur des informations, selon un **ordre** précis, et des **règles** bien définies, afin d'obtenir des informations mieux **exploitables**.

Definition

Informatique, concaténation du mot **information**, et traitement automatique, est la science de traitement automatique de l'information.

a

Definition

Un **Codage** est une **convention** pour changer la **représentation** des données, pour faciliter leur **conservation**, manipulation et leur transmission, en utilisant des lettres, chiffres et symboles.

Definition

Un **Codage** est une **convention** pour changer la **représentation** des données, pour faciliter leur **conservation**, manipulation et leur transmission, en utilisant des lettres, chiffres et symboles.

Exemple

Codage binaire : $(9)_{10} = 1001$

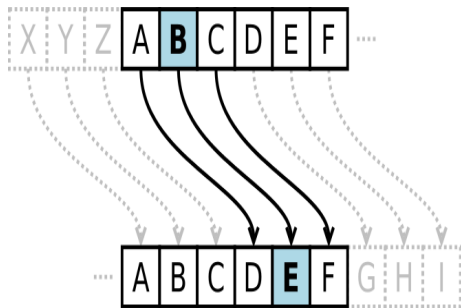
Definition

Un **Codage** est une **convention** pour changer la **représentation** des données, pour faciliter leur **conservation**, manipulation et leur transmission, en utilisant des lettres, chiffres et symboles.

Exemple

Codage binaire : $(9)_{10} = 1001$

Chiffre de César : Décalage de chaque lettre.



Definition

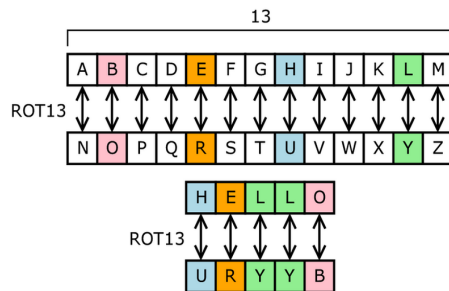
Un **Codage** est une **convention** pour changer la **représentation** des données, pour faciliter leur **conservation**, manipulation et leur transmission, en utilisant des lettres, chiffres et symboles.

Exemple

Codage binaire : $(9)_{10} = 1001$

Chiffre de César : Décalage de chaque lettre.

ROT13 : wikipedia :ROT 13



Definition

Un **Codage** est une **convention** pour changer la **représentation** des données, pour faciliter leur **conservation**, manipulation et leur transmission, en utilisant des lettres, chiffres et symboles.

Exemple

Codage binaire : $(9)_{10} = 1001$

Chiffre de César : Décalage de chaque lettre.

ROT13 : wikipedia :ROT 13

mp3 :

Definition

Un **Codage** est une **convention** pour changer la **représentation** des données, pour faciliter leur **conservation**, manipulation et leur transmission, en utilisant des lettres, chiffres et symboles.

Exemple

Codage binaire : $(9)_{|_2} = 1001$

Chiffre de César : Décalage de chaque lettre.

ROT13 : wikipedia :ROT 13

mp3 :

Avi,mgeg,...

Definition

Un entier A représenté dans une base β par une séquence de chiffres $a_i < \beta$:

$$(A)_{|\beta} = a_n a_{n-1} \dots a_1 a_0$$

Si :

$$A = \sum_{i=0}^n a_i \beta^i$$

1. $(410)_{10} =$

1. $(410)_{10} = 4 * 10^2 + 1 * 10^1 + 0 * 10^0 = 410$

2. $(1001101)_2 =$

$$1. (410)_{10} = 4 * 10^2 + 1 * 10^1 + 0 * 10^0 = 410$$

$$2. (1001101)_2 = 2^6 + 2^3 + 2^2 + 2^0 = 77$$

$$3. (8010)_8$$

1. $(410)_{10} = 4 * 10^2 + 1 * 10^1 + 0 * 10^0 = 410$

2. $(1001101)_2 = 2^6 + 2^3 + 2^2 + 2^0 = 77$

3. $(8010)_8$ **Ecriture fausse**

4. $(ABC2)_{16} =$

$$1. (410)_{10} = 4 * 10^2 + 1 * 10^1 + 0 * 10^0 = 410$$

$$2. (1001101)_2 = 2^6 + 2^3 + 2^2 + 2^0 = 77$$

$$3. (8010)_8 \quad \text{Ecriture fausse}$$

$$4. (ABC2)_{16} = 10 * 16^3 + 11 * 16^2 + 12 * 16^1 + 2 * 16^0 = 43969$$

$$\begin{array}{r|l} 2) \underline{15} & 1 \\ 2) \underline{7} & 1 \\ 2) \underline{3} & 1 \\ 2) \underline{1} & 1 \end{array} \left. \vphantom{\begin{array}{r} 15 \\ 7 \\ 3 \\ 1 \end{array}} \right\} 1111$$

15_{10}

15_{10}

$$\left. \begin{array}{r|l} 2)\underline{15} & 1 \\ 2)\underline{7} & 1 \\ 2)\underline{3} & 1 \\ 2)\underline{1} & 1 \end{array} \right\} 1111$$

 28_{10}

$$\left. \begin{array}{r|l} 2)\underline{28} & 0 \\ 2)\underline{14} & 0 \\ 2)\underline{7} & 1 \\ 2)\underline{3} & 1 \\ 2)\underline{1} & 1 \end{array} \right\} 11100$$

$$\begin{array}{r|l}
 15_{10} & \\
 2)\underline{15} & 1 \\
 2)\underline{7} & 1 \\
 2)\underline{3} & 1 \\
 2)\underline{1} & 1
 \end{array} \left. \vphantom{\begin{array}{r|l} 15_{10} \\ 2)\underline{15} \\ 2)\underline{7} \\ 2)\underline{3} \\ 2)\underline{1} \end{array}} \right\} 1111$$

$$\begin{array}{r|l}
 28_{10} & \\
 2)\underline{28} & 0 \\
 2)\underline{14} & 0 \\
 2)\underline{7} & 1 \\
 2)\underline{3} & 1 \\
 2)\underline{1} & 1
 \end{array} \left. \vphantom{\begin{array}{r|l} 28_{10} \\ 2)\underline{28} \\ 2)\underline{14} \\ 2)\underline{7} \\ 2)\underline{3} \\ 2)\underline{1} \end{array}} \right\} 11100$$

$$\begin{array}{r|l}
 156_{10} & \\
 2)\underline{156} & 0 \\
 2)\underline{78} & 0 \\
 2)\underline{39} & 1 \\
 2)\underline{19} & 1 \\
 2)\underline{9} & 1 \\
 2)\underline{4} & 0 \\
 2)\underline{2} & 0 \\
 2)\underline{1} & 1
 \end{array} \left. \vphantom{\begin{array}{r|l} 156_{10} \\ 2)\underline{156} \\ 2)\underline{78} \\ 2)\underline{39} \\ 2)\underline{19} \\ 2)\underline{9} \\ 2)\underline{4} \\ 2)\underline{2} \\ 2)\underline{1} \end{array}} \right\} 10011100$$

Exercice

Coder en base β les nombres suivants :

1. $(68)_{10} =$

2. $(423)_{10} =$

3. $(137)_{10} =$

4. $(450)_{10} =$

Exercice

Coder en base β les nombres suivants :

1. $(68)_{10} = 1000100$

2. $(423)_{10} =$

3. $(137)_{10} =$

4. $(450)_{10} =$

Exercice

Coder en base β les nombres suivants :

1. $(68)_{10} = 1000100$

2. $(423)_{10} = 110100111$

3. $(137)_{10} =$

4. $(450)_{10} =$

Exercice

Coder en base β les nombres suivants :

1. $(68)_{10} = 1000100$
2. $(423)_{10} = 110100111$
3. $(137)_{10} = 211$
4. $(450)_{10} =$

Exercice

Coder en base β les nombres suivants :

1. $(68)_{10} = 1000100$
2. $(423)_{10} = 110100111$
3. $(137)_{10} = 211$
4. $(450)_{10} = 1C2$

Conversion

Pour passer d'une représentation d'un nombre dans une base β_1 à une autre base β_2 . On peut :

- ▶ Calculer la valeur dans la base décimale.
- ▶ Coder la valeur trouvée dans la base β_2 .

Exemple

1. $(467)_{10} =$

Conversion

Pour passer d'une représentation d'un nombre dans une base β_1 à une autre base β_2 . On peut :

- ▶ Calculer la valeur dans la base décimale.
- ▶ Coder la valeur trouvée dans la base β_2 .

Exemple

1. $(467)_{|8} = 311 = (100110111)_{|2}$.

2. $(34B)_{|16} =$

Conversion

Pour passer d'une représentation d'un nombre dans une base β_1 à une autre base β_2 . On peut :

- ▶ Calculer la valeur dans la base décimale.
- ▶ Coder la valeur trouvée dans la base β_2 .

Exemple

1. $(467)_{10} = 311 = (100110111)_2$.

2. $(34B)_{16} = 843 = (1101001011)_2$

Definition

Il existe plusieurs méthodes pour coder un nombre **négatif** en binaire. La méthode la plus répandue est **Complément à deux**.
Pour cette méthode on doit **fixer** le nombre de bits réservés pour le codage.

Definition

Il existe plusieurs méthodes pour coder un nombre **négatif** en binaire. La méthode la plus répandue est **Complément à deux**.

Pour cette méthode on doit **fixer** le nombre de bits réservés pour le codage.

Ainsi pour coder un nombre $-A$ avec $A \geq 0$ en n bits. On effectue les opérations suivantes :

1. **Coder** A sur n bits.

$$-21 =$$

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

Definition

Il existe plusieurs méthodes pour coder un nombre **négatif** en binaire. La méthode la plus répandue est **Complément à deux**.

Pour cette méthode on doit **fixer** le nombre de bits réservés pour le codage.

Ainsi pour coder un nombre $-A$ avec $A \geq 0$ en n bits. On effectue les opérations suivantes :

1. **Coder** A sur n bits.
2. **Complément à un**, une opération qui consiste à inverser chaque bit du codage.

$$-21 =$$

0	0	0	1	0	1	0	1
1	1	1	0	1	0	1	0

Definition

Il existe plusieurs méthodes pour coder un nombre **négatif** en binaire. La méthode la plus répandue est **Complément à deux**.

Pour cette méthode on doit **fixer** le nombre de bits réservés pour le codage.

Ainsi pour coder un nombre $-A$ avec $A \geq 0$ en n bits. On effectue les opérations suivantes :

1. **Coder** A sur n bits.
2. **Complément à un**, une opération qui consiste à inverser chaque bit du codage.
3. **Ajouter 1** au code obtenu.

$$-21 =$$

0	0	0	1	0	1	0	1
1	1	1	0	1	0	1	0
1	1	1	0	1	0	1	1

Definition

Pour le codage des nombres **réels**. On code la partie **entière** en base 2, et la partie réelle en base 2^{-1} .

Definition

Pour le codage des nombres **réels**. On code la partie **entière** en base 2, et la partie réelle en base 2^{-1} .

Exemple

Convertir le nombre **6.625** en base binaire.

1. Coder la partie entière :

$$6_{|2} = 110$$

2. Coder la partie réelle par des **multiplications** par 2.

$$0.625_{|2} = 0.101$$

3. finalement :

$$6.625_{|2} = 110.101$$

Algorithmique

Définition

Un **Algorithme** est une suite **d'opérations** de calcul élémentaires, organisée selon des règles précises, dont le but est de **résoudre** un problème bien défini.

Les **opérations** d'un algorithme doivent être simples et directes :

Définition

Un **Algorithme** est une suite **d'opérations** de calcul élémentaires, organisée selon des règles précises, dont le but est de **résoudre** un problème bien défini.

Les **opérations** d'un algorithme doivent être simples et directes :

- ▶ Opérations arithmétique : $+$, $-$, $/$, $*$, \dots

Définition

Un **Algorithme** est une suite d'**opérations** de calcul élémentaires, organisée selon des règles précises, dont le but est de résoudre un problème bien défini.

Les **opérations** d'un algorithme doivent être simples et directes :

- ▶ Opérations arithmétique : $+$, $-$, $/$, $*$, \dots
- ▶ Opérations logiques : $a < b$, $a = b$, P *ou* Q .

Définition

Un **Algorithme** est une suite **d'opérations** de calcul élémentaires, organisée selon des règles précises, dont le but est de **résoudre** un problème bien défini.

Les **opérations** d'un algorithme doivent être simples et directes :

- ▶ Opérations arithmétique : $+$, $-$, $/$, $*$, \dots
- ▶ Opérations logiques : $a < b$, $a = b$, P *ou* Q .
- ▶ **transfert** des données : **lecture** et **écriture**.

Algorithmique

Algorithmique

- ▶ L'étude des **algorithmes** et des problèmes qui s'y attachent a donné lieu à une branche en informatique : **ALGORITHMIQUE**.

Algorithmique

- ▶ L'étude des **algorithmes** et des problèmes qui s'y attachent a donné lieu à une branche en informatique : **ALGORITHMIQUE**.
- ▶ **l'algorithmique** est un **ensemble de règles** et des **techniques** qui sont impliqués dans la définition et la conception d'algorithmes efficaces.

Algorithmique

- ▶ L'étude des **algorithmes** et des problèmes qui s'y attachent a donné lieu à une branche en informatique : **ALGORITHMIQUE**.
- ▶ **l'algorithmique** est un **ensemble de règles** et des **techniques** qui sont impliqués dans la définition et la conception d'algorithmes efficaces.
- ▶ **l'algorithmique** est aussi une science qui étudie **l'efficacité** ou la **rapidité** des algorithmes en regardant le nombre d'opérations, ou le temps de calcul qu'il nécessite.

Exemple d'algorithme

Exemple d'algorithme

1. Calculer la représentation en *binaire* d'un entier naturel.

Exemple d'algorithme

1. Calculer la représentation en *binaire* d'un entier naturel.
2. Calculer les racines d'un polynôme de deuxième degré.

Exemple d'algorithme

1. Calculer la représentation en *binaire* d'un entier naturel.
2. Calculer les racines d'un polynôme de deuxième degré.
3. Calculer le plus grand diviseur commun de deux entiers.

Exemple d'algorithme

1. Calculer la représentation en *binaire* d'un entier naturel.
2. Calculer les racines d'un polynôme de deuxième degré.
3. Calculer le plus grand diviseur commun de deux entiers.
4. Multiplication de deux matrices.

Exemple d'algorithme

1. Calculer la représentation en *binaire* d'un entier naturel.
2. Calculer les racines d'un polynôme de deuxième degré.
3. Calculer le plus grand diviseur commun de deux entiers.
4. Multiplication de deux matrices.
5. Trouver le chemin le plus court dans un graphe complexe.



Definition

Pour un vecteur V de \mathbb{R}^n , un sommet à la position i vérifie :

$$V_{i-1} \leq V_i \geq V_{i+1}$$

avec la **convention** que $V_{-1} = -\infty$ et $V_n = -\infty$

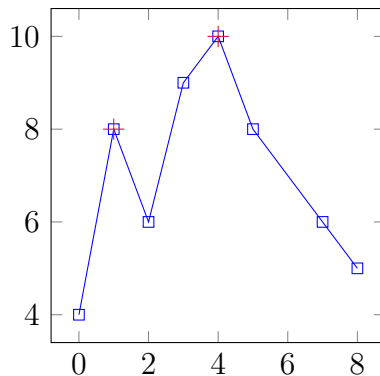
Definition

Pour un vecteur V de \mathbb{R}^n , un sommet à la position i vérifie :

$$V_{i-1} \leq V_i \geq V_{i+1}$$

avec la **convention** que $V_{-1} = -\infty$ et $V_n = -\infty$

$-\infty$	4	8	6	9	10	8	7	6	5	$-\infty$
	0	1	2	3	4	5	6	7	8	



Algorithme 1 : Force brute

Données : V vecteur de \mathbb{R}^n

```
1 for  $i=0$  à  $n-1$  do
2   |   if  $V_{i-1} \leq V_i \geq V_{i+1}$  then
3   |   |   Sommet  $\leftarrow V_i$ 
4   |   end
5 end
```

Résultat : Sommet

Algorithme 2 : Recherche du max

Données : V vecteur de \mathbb{R}^n

```
1  $Max \leftarrow -\infty$ 
2 for  $i=0$  à  $n-1$  do
3   |   if  $V_i > Max$  then
4   |   |   /* nouveau maximum */
4   |   |    $Max \leftarrow V_i$ 
5   |   end
6 end
```

Résultat : Max

Algorithme 3 : Diviser pour régner

```
1 RechSommet( $A, i, j$ ) :  
    /* Calculer le milieu */  
2    $mid = \frac{i + j}{2}$   
3   if ( $A_{mid-1} \leq A_{mid} \geq A_{mid+1}$ ) then  
4   |   Return  $A_{mid}$   
5   else if  $A_{mid} \leq A_{mid+1}$  then  
6   |   /* Sommet à droite */  
7   |   Return RechSommet( $A, mid + 1, j$ )  
8   else  
9   |   /* Sommet à gauche */  
10  |   Return RechSommet( $A, i, mid - 1$ )  
11 end
```

Définition

Un **type** en algorithmique est une information permettant de **traduire** les valeurs depuis une représentation binaire (celle de l'ordinateur) vers une autre représentation plus **adaptée**.

Définition

Un **type** en algorithmique est une information permettant de **traduire** les valeurs depuis une représentation binaire (celle de l'ordinateur) vers une autre représentation plus **adaptée**.

Caractéristiques

Un **type** est caractérisé par :

- ▶ un nom.
- ▶ un ensemble de valeurs.
- ▶ un ensemble d'opérations définis sur ces valeurs.

Nous distinguons **quatre** type élémentaires en algorithmique :

entier : représente **une partie** de \mathbb{Z} , avec les opérations arithmétiques simples :

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

Nous distinguons **quatre** type élémentaires en algorithmique :

entier : représente **une partie** de \mathbb{Z} , avec les opérations arithmétiques simples :

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

Réel : utilisé pour représenter une **partie** de \mathbb{R} muni des mêmes opérations que les **entiers**.

Nous distinguons **quatre** type élémentaires en algorithmique :

entier : représente **une partie** de \mathbb{Z} , avec les opérations arithmétiques simples :

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

Réel : utilisé pour représenter une **partie** de \mathbb{R} muni des mêmes opérations que les **entiers**.

Caractère : représente un **caractère** imprimable, et seront écrits entre : $\{ ' , " \}$

Nous distinguons **quatre** type élémentaires en algorithmique :

entier : représente **une partie** de \mathbb{Z} , avec les opérations arithmétiques simples :

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

Réel : utilisé pour représenter une **partie** de \mathbb{R} muni des mêmes opérations que les **entiers**.

Caractère : représente un **caractère** imprimable, et seront écrits entre : $\{ ' , " \}$

booléen : sera utilisé pour stocker les valeurs de **Vrai,Faux**, menu par les opérations logiques : **ou,et,non**.

Définition

Une **Variable** est une **donnée** qu'un programme peut **manipuler**.

Toute variable possède :

Définition

Une **Variable** est une **donnée** qu'un programme peut **manipuler**.

Toute variable possède :

type : précisant sa nature : (entier,réel,caractère,booléen).

Définition

Une **Variable** est une **donnée** qu'un programme peut **manipuler**.

Toute variable possède :

type : précisant sa nature : (entier,réel,caractère,booléen).

nom : Un **identificateur** que l'utilisateur choisit pour manipuler cette variable.

Définition

Une **Variable** est une **donnée** qu'un programme peut **manipuler**.

Toute variable possède :

type : précisant sa nature : (entier,réel,caractère,booléen).

nom : Un **identificateur** que l'utilisateur choisit pour manipuler cette variable.

valeur : qui peut **évoluer** au cours du programme,mais doit **respecter** le **type**.

Dans un algorithme, toutes les variables utilisées doivent être **déclarées**.

Dans un algorithme, toutes les variables utilisées doivent être **déclarées**.

Déclaration

Une variable est déclarée au début d'un algorithme en précédant son nom par **VAR**, puis on précise son type après ' : '.

Exemple

Var	salaire	:	Entier.
Var	taux	:	Réel.
Var	racine1, racine2	:	Réel.
Var	testDiscriminant	:	Booléen

Definition

L'opération d'affectation est une **attribution** d'une valeur à une variable. elle se note par le signe ←

Definition

L'opération d'affectation est une **attribution** d'une valeur à une variable. elle se note par le signe ←

Definition

L'opération d'affectation est une **attribution** d'une valeur à une variable. elle se note par le signe \leftarrow

-
-
- 1 **Var A, B : Entier**
 - 2 **Var C : Caractère**
 - 3 $A \leftarrow 1$ Affecte la valeur de 1 à A.
 - 4 $B \leftarrow A$ {donne à B la valeur de A, A ne change pas}
 - 5 $A \leftarrow A + 1$ augment la valeur de A par 1.
 - 6 $c = '2'$ un caractère.
 - 7 $c \leftarrow A$ erreur, car A et c ne sont pas de même types.
-

Donner les valeurs des variables déclarées à la fin de chaque algorithme.

Algorithme 4 : Affectation 1

1 **Var A : Entier.**

2 $A \leftarrow 13$

3 $B \leftarrow 7$

4 $B \leftarrow A - B$

5 $A \leftarrow A/2$

6 $B \leftarrow A + 1$

Algorithme 5 : Affectation 2

1 **Var A : Entier.**

2 **Var x : Réel.**

3 **Var test : Réel**

4 $A \leftarrow 13$

5 $x \leftarrow 7.5$

6 $A \leftarrow A + x$

7 $x \leftarrow x/2$

8 $x \leftarrow 2(A + x)$

9 $\text{test} \leftarrow x > A$

Les instructions de lecture et écriture(**Entrée/sortie**) permettent au programmeur de communiquer avec l'utilisateur.

Definition

La **lecture** est l'opération de base qui consiste à lire des données tapées au clavier.

Les instructions de lecture et écriture(**Entrée/sortie**) permettent au programmeur de communiquer avec l'utilisateur.

Definition

La **lecture** est l'opération de base qui consiste à lire des données tapées au clavier.

Les instructions de lecture et écriture(**Entrée/sortie**) permettent au programmeur de communiquer avec l'utilisateur.

Definition

La **lecture** est l'opération de base qui consiste à lire des données tapées au clavier.

Lire(Salaire)

Est une **instruction** qui permet à l'utilisateur d'entrer une **nouvelle valeur** pour la variable salaire. à son exécution le programme **s'interrompt**, pour attendre la saisie de cette valeur.

- L'écriture est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le contenu de la variable Var.

- L'écriture est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le contenu de la variable Var.

- ▶ **L'écriture** est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le **contenu** de la variable Var.

- ▶ ▶ **Var** a,b : Entier

- ▶ **L'écriture** est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le **contenu** de la variable Var.

- ▶ **Var** a,b : Entier
- ▶ **Var** C : Caractère

- ▶ **L'écriture** est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le **contenu** de la variable Var.

- ▶
 - ▷ **Var** a,b : Entier
 - ▷ **Var** C : Caractère
 - ▷ $a \leftarrow 4$ $b \leftarrow 5$ $C \leftarrow \text{'somme'}$

- ▶ **L'écriture** est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le **contenu** de la variable Var.

- ▶
 - ▷ **Var** a,b : Entier
 - ▷ **Var** C : Caractère
 - ▷ $a \leftarrow 4$ $b \leftarrow 5$ $C \leftarrow \text{'somme'}$
 - ▷ **Ecrire**(a)

- ▶ **L'écriture** est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le **contenu** de la variable Var.

- ▶
 - ▷ **Var** a,b : Entier
 - ▷ **Var** C : Caractère
 - ▷ $a \leftarrow 4 \quad b \leftarrow 5 \quad C \leftarrow \text{'somme'}$
 - ▷ **Ecrire**(a)
 - ▷ **Ecrire**('b=',b)

- ▶ **L'écriture** est une instruction permettant à l'algorithme (ou programmeur) de communiquer (ou afficher) les résultats des valeurs calculées à l'utilisateur.

Ecrire(Var)

Affichera le **contenu** de la variable Var.

- ▶
 - ▷ **Var** a,b : Entier
 - ▷ **Var** C : Caractère
 - ▷ $a \leftarrow 4$ $b \leftarrow 5$ $C \leftarrow \text{'somme'}$
 - ▷ **Ecrire**(a)
 - ▷ **Ecrire**('b=',b)
 - ▷ **Ecrire**(C,a+b)

Voici un exemple d'un algorithme qui demande une température en **Celsius**, et la convertis en **Fahrenheit**.

Voici un exemple d'un algorithme qui demande une température en **Celsius**, et la convertis en **Fahrenheit**.

Algorithme 7 : Conversion Celsius à Fahrenheit

```
1 Var C,F : Réel.  
2 {LECTURE}  
3 Ecrire( 'donner une température en Celsius : ');  
4 Lire(C)  
5 {TRAITEMENT}  
6  $F \leftarrow 1.8 C + 32$   
7 {AFFICHAGE}  
8 Ecrire(C, 'C =', F, 'F')
```

Moyenne

Ecrire un **algorithme** qui calcule la **moyenne** arithmétique et géométrique de deux entiers.

Moyenne

Ecrire un **algorithme** qui calcule la **moyenne** arithmétique et géométrique de deux entiers.

Algorithme 9 : Moyennes de deux entiers

```
1 Var a,b : Entier.  
2 Var Ma,Mg : Réel.  
  
3 Ecrire('Donner deux entiers : ');  
4 Lire(a)  
5 Lire(b)  
  
6  $Ma \leftarrow (a + b)/2$   
7  $Mg \leftarrow \sqrt{ab}$   
  
8 Ecrire('Moyenne arithmetique = ',Ma)  
9 Ecrire('Moyenne géométrique = ',Mg)
```

Permutation

Ecrire un algorithme qui lit deux **réels**, puis permute leur valeurs.

Permutation

Ecrire un algorithme qui lit deux réels, puis permute leur valeurs.

Algorithme 11 : Permutation de deux valeurs

```
1 Var x,y, tmp : Réel.  
2 Ecrire('Donner deux réels : ');  
3 Lire(a)  
4 Lire(b)  
5  $tmp \leftarrow x$   
6  $x \leftarrow y$   
7  $y \leftarrow tmp$   
8 Ecrire('x= ',x)  
9 Ecrire('y= ',y)
```

pièce de monnaie

Donner un **algorithme** qui reçoit une **somme d'argent** ($< 50DH$) puis l'affiche en forme de pièces de monnaies **élémentaires**, d'une manière optimisée.

Exemple

34DH=

pièce de monnaie

Donner un **algorithme** qui reçoit une **somme d'argent** ($< 50DH$) puis l'affiche en forme de pièces de monnaies **élémentaires**, d'une manière optimisée.

Exemple

$$34DH = 1 * 20DH + 1 * 10DH + 2 * 2DH$$

pièce de monnaie

Donner un **algorithme** qui reçoit une **somme d'argent** ($< 50\text{DH}$) puis l'affiche en forme de pièces de monnaies **élémentaires**, d'une manière optimisée.

Exemple

$34\text{DH} = 1 * 20\text{DH} + 1 * 10\text{DH} + 2 * 2\text{DH}$

Algorithme 14 : Décomposition

```

1  Var somme,q : Entier.

2  Ecrire('Entrer une Somme (< 50) : ');
3  Lire(somme)

4  somme,Dh=
5   $q \leftarrow \frac{\text{Somme}}{20}$ 
6  ecrire(q,'*20DH +')
7   $\text{somme} \leftarrow \text{somme} - 20 * q$ 

8   $q \leftarrow \frac{\text{Somme}}{10}$ 
9  ecrire(q,'*10DH +')
10  $\text{somme} \leftarrow \text{somme} - 10 * q$ 

11  $q \leftarrow \frac{\text{Somme}}{5}$ 
12 ecrire(q,'*5DH +')
13  $\text{somme} \leftarrow \text{somme} - 5 * q$ 
14 ecrire(somme,'*1DH')
```