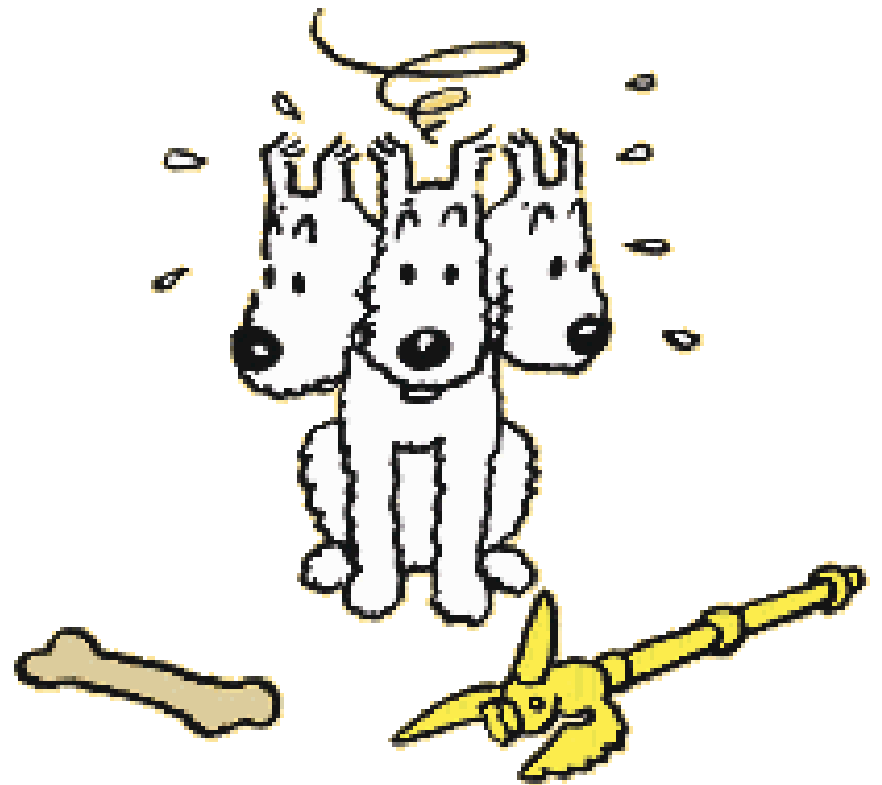
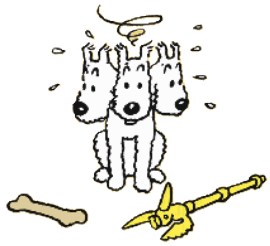


# Logique de programmation

Structure de contrôle : l'alternative





# Alternative simple

---

La structure alternative permet d'effectuer une suite d'instructions si une condition est remplie et d'en effectuer une autre si celle-ci ne l'est pas.

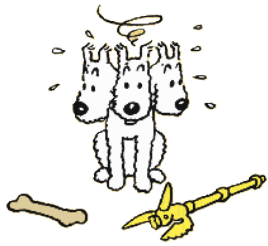
L'exécution de l'alternative commence par l'évaluation de la condition (vraie ou fausse) suivie de l'exécution de la suite d'instructions associées à la réponse obtenue.

Syntaxe de la structure "alternative simple":

```
SI expressionBooléen  
  ALORS  
    [0..n] Instructions  
  SINON  
    [0..n] Instructions  
FINSI
```

Lorsque le résultat de l'évaluation de l'expression booléenne est :

- **VRAI** : seules les instructions entre le **ALORS** et le **SINON** sont exécutées
- **FAUX** : seules les instructions entre le **SINON** et le **FINSI** sont exécutées

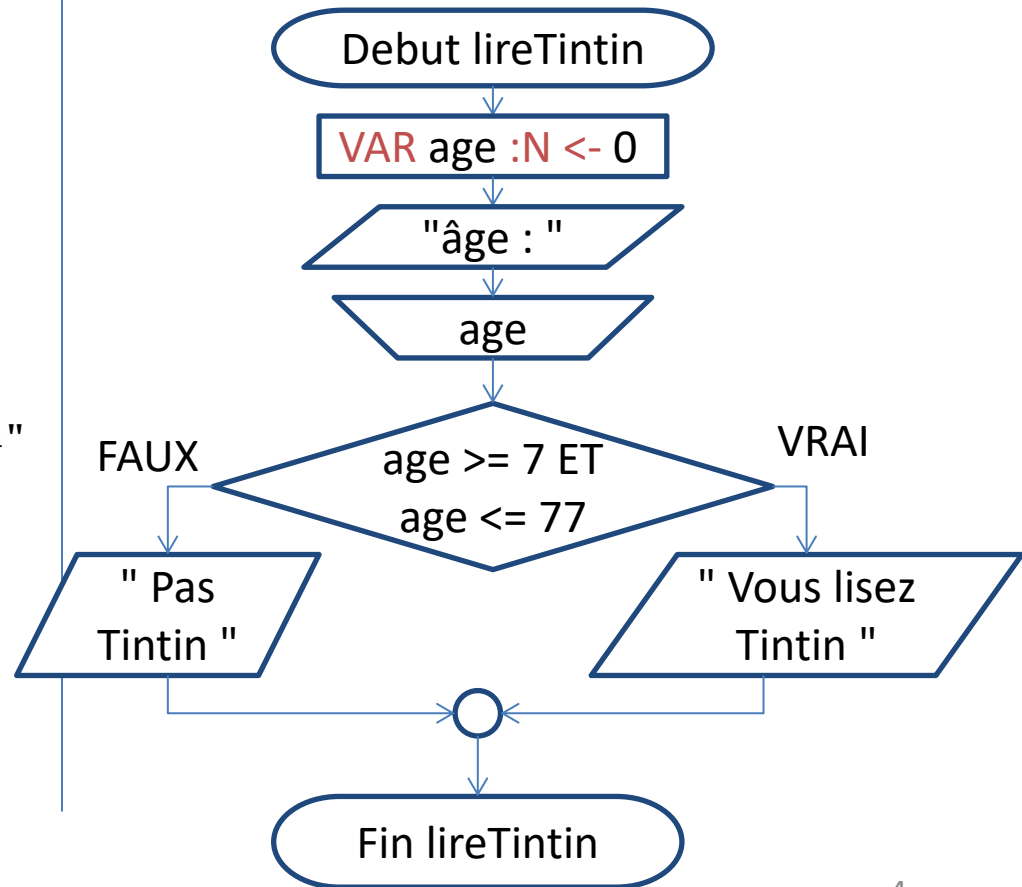


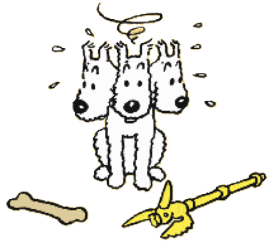
# Représentation

## Pseudo - code

```
1. Debut lireTintin
2.  VARIABLES LOCALES :
3.    VAR age :N <- 0
4.  FIN VARIABLES LOCALES
5.  Afficher "âge : "
6.  Saisir age
7.  SI age >= 7 ET age <= 77
8.    ALORS
9.      Afficher "vous lisez Tintin"
10.   SINON
11.     Afficher " pas de Tintin"
12.   FINSI
13. Fin lireTintin
```

## Ordinogramme



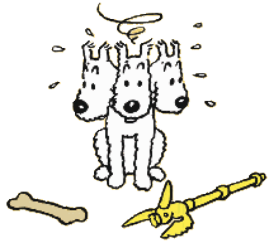


# Table de valeurs

```
1. Debut lireTintin
2. VARIABLES LOCALES :
3.   VAR age :N <- 0
4. FIN VARIABLES LOCALES
5. Afficher "âge : "
6. Saisir age
7. SI age >= 7 ET age <= 77
8.   ALORS
9.     Afficher
       "vous lisez Tintin"
10. SINON
11.   Afficher
       " pas de Tintin"
12. FINSI
13. Fin lireTintin
```

N°	age	Ecran / clavier
1	?	
2	?	
3	0	
4	0	
5	0	-> âge :
6	80	<- 80
7	80	
10	80	
11	80	->pas de Tintin
12	80	
13	?	

N°	age	Ecran / clavier
1	?	
2	?	
3	0	
4	0	
5	0	-> âge :
6	35	<- 35
7	35	
8	35	
9	35	->vous lisez Tintin
12	35	
13	?	



# Alternative simple variante

- Lorsque aucune instruction n'est à exécuter quand le test est faux, on n'indiquera pas le sinon.

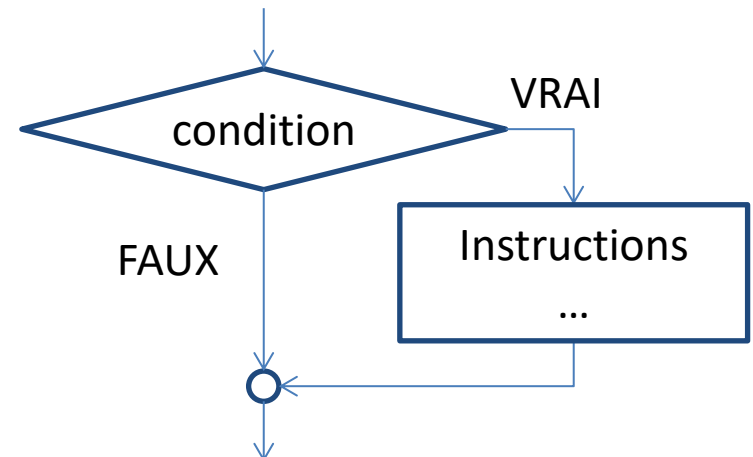
Syntaxe de la structure "alternative simple variante":

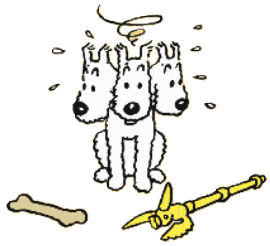
```
SI experssionBooléen  
  ALORS  
    [1..n] Instructions  
FINSI
```

## Pseudo - code

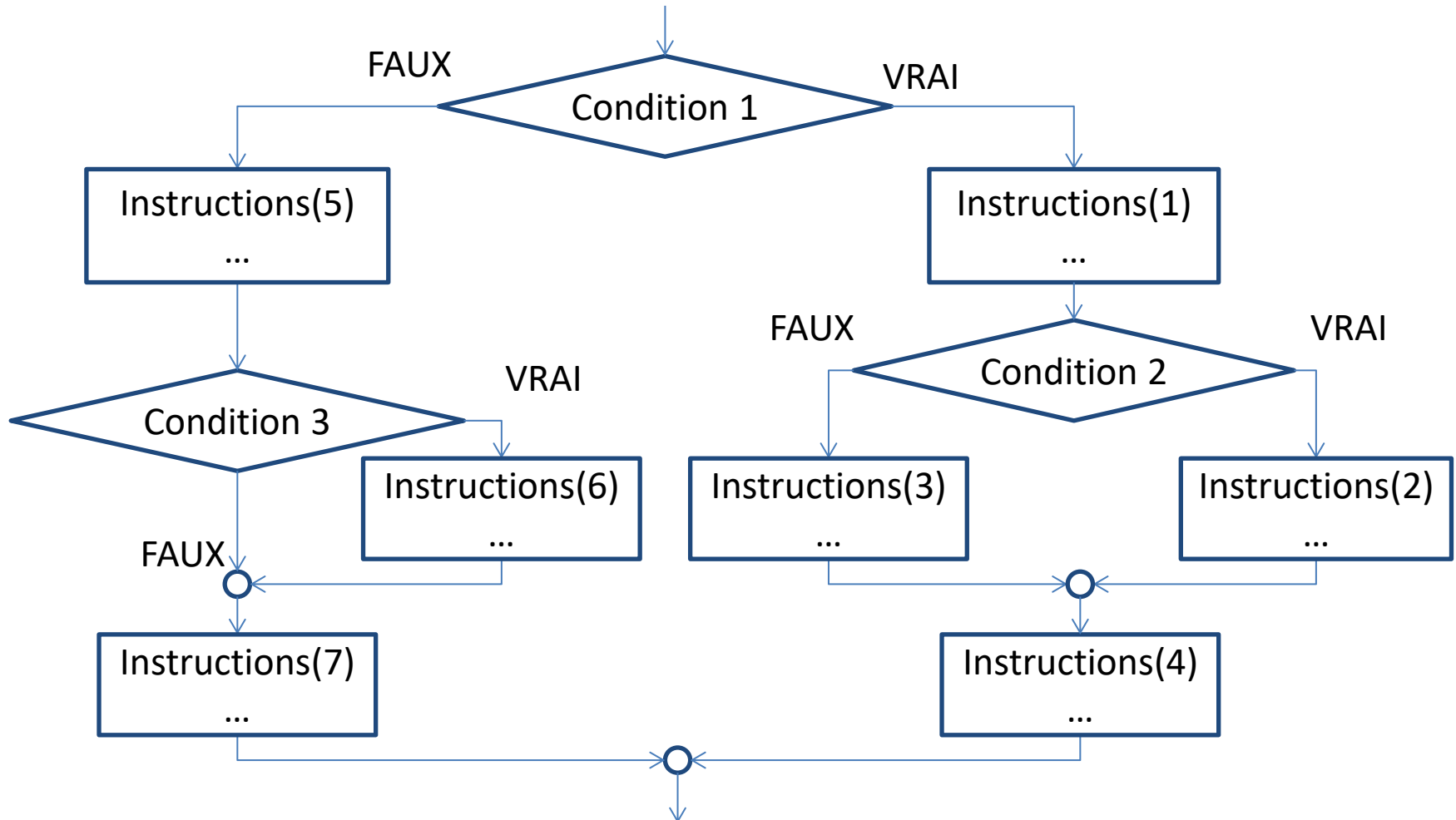
```
SI condition  
  ALORS  
    Instruction  
  ...  
FINSI
```

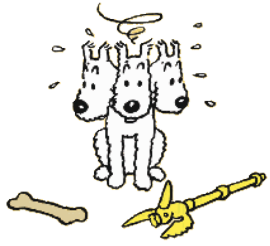
## Ordinogramme





# Alternative simple imbriquée



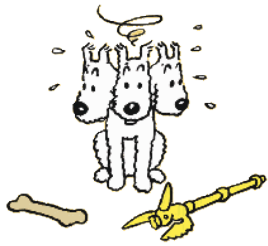


# Alternative simple imbriquée (pseudo-code)

---

```
SI condition 1
  ALORS
    instructions (1)
  SI condition 2
    ALORS
      instructions (2)
    SINON
      instructions (3)
  FINSI
  instructions (4)
SINON
  instructions (5)
  SI condition 3
    ALORS
      instructions (6)
  FINSI
  instructions (7)
FINSI
```





# Alternative composée

La structure alternative composée permet d'effectuer une suite d'instructions en fonction de la valeur d'une variable.

L'exécution de l'alternative commence par une recherche du bloc d'instructions liées à la valeur de la variable et se poursuit en exécutant les instructions de ce bloc.

Syntaxe de la structure "alternative composée":

```
CAS OU expressionNumérique
  CAS valeurNumérique
    [0..n] Instructions
  CAS autreValeurNumérique
    [0..n] Instructions

  ...

  AUTRES CAS
    [0..n] Instructions
FIN CAS OU
```

- Seules les instructions se trouvant dans le cas qui a la même valeur que l'expression numérique seront exécutées.
- Si aucun cas n'a la valeur de l'expression, ce sont les instructions de "AUTRE CAS" qui sont exécutées.

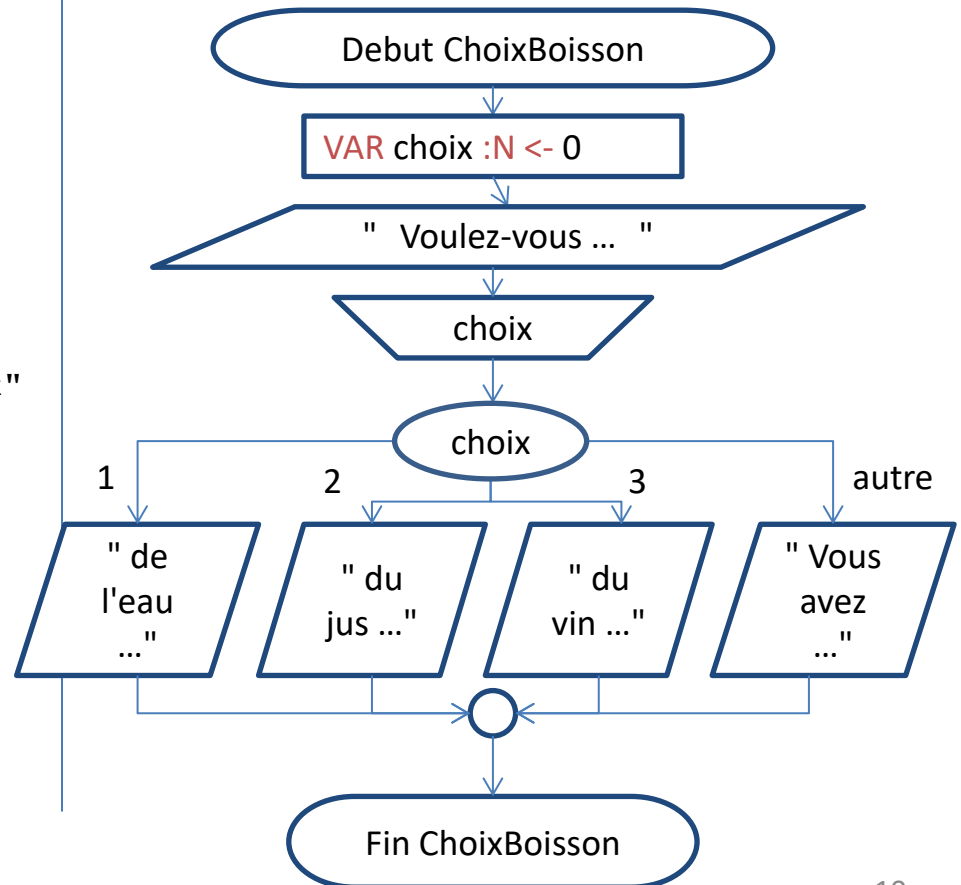


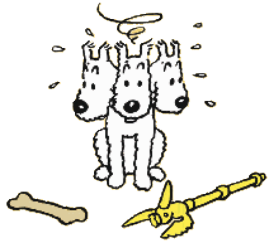
# Représentation

## Pseudo - code

```
Debut ChoixBoisson
  VARIABLES LOCALES :
    VAR choix:N <- 0
  FIN VARIABLES LOCALES
  Afficher "Voulez-vous : 1 de l'eau, 2
    du jus, 3 du vin"
  Saisir choix
  CAS OU choix
    CAS 1
      Afficher "de l'eau très bon choix"
    CAS 2
      Afficher "du jus bravo"
    CAS 3
      Afficher "du vin choix judicieux"
    AUTRE CAS
      Afficher "Vous avez raison de ne
        pas boire"
    FIN CAS OU
  Fin ChoixBoisson
```

## Ordinogramme





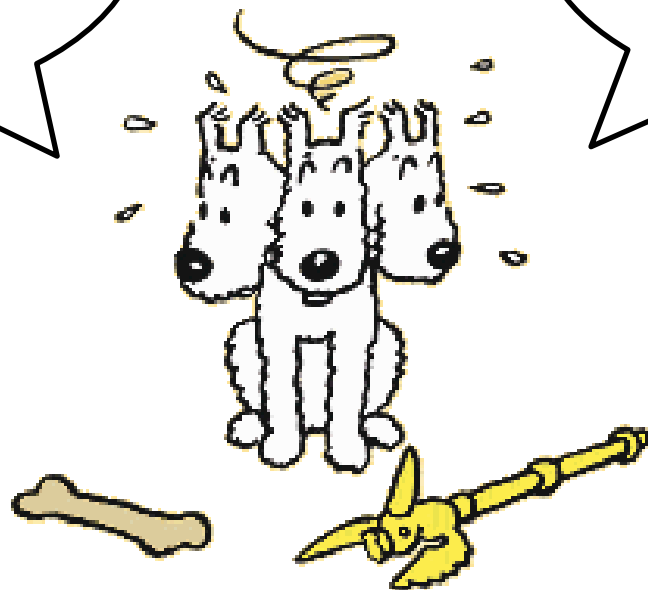
# Synthèse

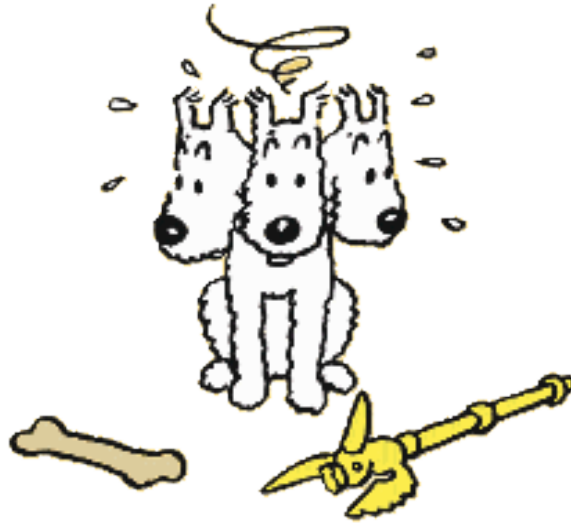
---

- Alternative simple
  - Représentation
  - Table de valeurs
  - Variante sans "sinon"
  - Imbrication d'alternatives
  - Exercices
- Alternative Composée
  - Exercices

Merci

A vous  
maintenant





Structure de contrôle : l'alternative

# EXERCICES

### Début et fin de programme

**DEBUT** nomDuProgramme

**FIN** nomDuProgramme

### Variables et constantes

**VARIABLES LOCALES :**

[n]CONST Nom : Type <- Littéral

[n]VAR Nom : Type <- Littéral

**FIN VARIABLES LOCALES**

### Instruction d'assignation

nomVariable <- Expression

### Affichage de données

**AFFICHER** expression1, ...

### Saisie de données

**SAISIR** variable

### Alternative simple

**SI** experssionBooléen

**ALORS**

[n] Instructions

**SINON**

[n] Instructions

**FINSI**

### Alternative composée

**CAS OU** experssionNumérique

**CAS** valeurNumérique

[n] Instructions

...

**AUTRES CAS**

[n] Instructions

**FIN CAS OU**

## Machine logique



### Evaluation des expressions

1. Chaque variable est remplacée par sa valeur actuelle
2. Chaque opération est effectuée puis remplacée par son résultat dans l'ordre de priorité des opérateurs

### Expression

- Littéral
- Constante ou variable
- Opération

### Données Numériques

**Symbole : N**

**Littéraux :** 12 456,7 0,005

**Opérateurs :**

( ), <sup>a</sup>, <sup>b</sup>, va, +, -, \*, /, DIV, MOD, +, -

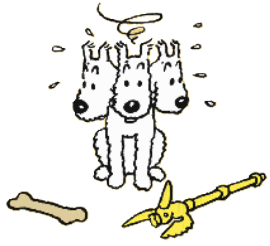
### Données booléennes

**Symbole : B**

**Littéraux :** VRAI FAUX

**Opérateurs :**

( ), Non, <, ≤, ≥, =, ≠, et, ou

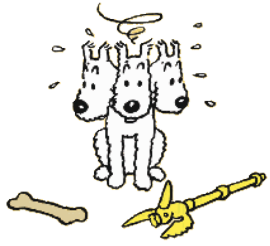


## Exercice 3.1 : Equation 1<sup>er</sup> degré

---

- Écrire le programme qui résout l'équation du 1<sup>er</sup> degré:  $ax + b = 0$

Le programme demande à l'utilisateur une valeur pour a et pour b et donne si possible la valeur de x.

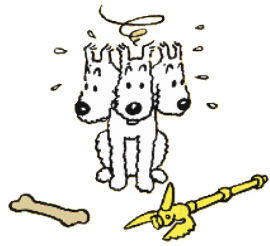


## Exercice 3.2 : Comparaison de 2 nombres

---

Écrire le programme qui compare deux nombres et qui indique quel est le plus grand ou s'ils sont égaux.





## Exercice 3.3 : Décomposition d'une somme d'argent

---

Écrire le programme qui permet de décomposer une somme d'argent lue au clavier.

Coupures disponibles :

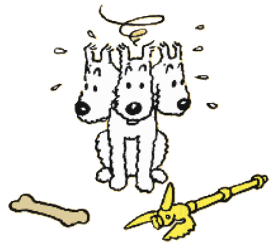
- Billets : 100€, 50€, 20€, 5€
- Pièces : 2€, 1€, 0,5€, 0,2€

Contraintes :

- Le total de la décomposition sera plus grand ou égale à la somme entrée
- Le nombre de coupures de la décomposition sera le plus petite possible

Exemple:

- 254,18 € se décompose en :
  - 2 billets de 100 €,
  - 1 billet de 50 €,
  - 2 pièces de 2 €,
  - 1 pièce de 0,2 €.



## Exercice 3.4 : Comparaison de 3 nombres

---

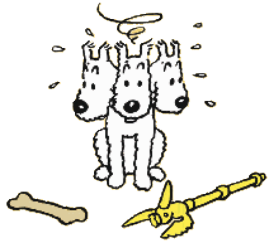
Écrire le programme qui compare 3 nombres entre eux.

Tenir compte d'une égalité possible des nombres.

### **Attention :**

«  $5 < 8 > 4$  » n'indique pas la comparaison entre le 1<sup>er</sup> et le 3<sup>ème</sup> nombre.

On préférera par exemple «  $4 < 5 < 8$  ».



## Exercice 3.5 : TVA

---

Ecrire un programme qui demande à l'utilisateur de choisir un taux de TVA et d'entrer un prix(hors TVA). Le programme affiche le taux choisi, le montant de la TVA et le prix TVA comprise.

Tva 6% (1)

Tva 12% (2)

Tva 21% (3)

Tva 0 % (4)

Sortie (0)

Votre Choix : \_\_\_\_\_

Prix (hors TVA) : \_\_\_\_\_

Le taux de TVA vaut : XXX

Le montant de la TVA est : XXX

Le prix T.V.A.C vaut : XXX

- **3.1 : Equation 1<sup>er</sup> degré**
  - Écrire le programme qui résout l'équation du 1<sup>er</sup> degré:  $ax + b = 0$   
Le programme demande à l'utilisateur une valeur pour a et pour b et donne si possible la valeur de x.
- **Exercice 3.2 : Comparaison de 2 nombres**

Écrire le programme qui compare deux nombres et qui indique quel est le plus grand ou s'ils sont égaux.
- **Exercice 3.3 : Décomposition d'une somme d'argent**
  - Écrire le programme qui permet de décomposer une somme d'argent lue au clavier.  
Coupures disponibles :
    - Billets : 100€, 50€, 20€, 5€
    - Pièces : 2€, 1€, 0,5€ , 0,2€
Contraintes :
    - Le total de la décomposition sera plus grand ou égale à la somme entrée
    - Le nombre de coupures de la décomposition sera le plus petite possible
Exemple:
    - 254,18 € se décompose en :
      - » 2 billets de 100 €,
      - » 1 billet de 50 €,
      - » 2 pièces de 2 €,
      - » 1 pièce de 0,2 €.
- **Exercice 3.4 : Comparaison de 3 nombres**
  - Écrire le programme qui compare 3 nombres entre eux.  
Tenir compte d'une égalité possible des nombres.  
**Attention :**  
« 5 < 8 > 4 » n'indique pas la comparaison entre le 1<sup>er</sup> et le 3<sup>ème</sup> nombre.  
On préférera par exemple « 4 < 5 < 8 ».
- **Exercice 3.5 : TVA**
  - Ecrire un programme qui demande à l'utilisateur de choisir un taux de TVA et d'entrer un prix(hors TVA). Le programme affiche le taux choisi, le montant de la TVA et le prix TVA comprise.  
Tva 6% (1)  
Tva 12% (2)  
Tva 21% (3)  
Tva 0 % (4)  
Sortie (0)  
Votre Choix : \_\_\_\_\_  
Prix (hors TVA) : \_\_\_\_\_  
Le taux de TVA vaut : XXX  
Le montant de la TVA est : XXX  
Le prix T.V.A.C vaut : XXX