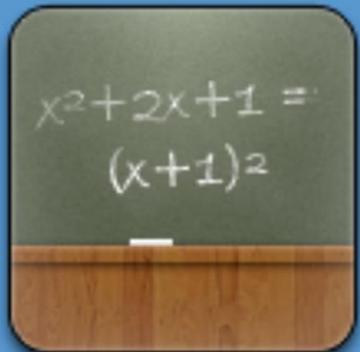




Programmation



Cursus
1

Level
1

Algorithmie

Module 1 Partie 2





Objectifs

- Aborder la programmation.
- Présentation d'un langage universel.
- Organiser un programme.
- Utiliser des données élémentaires.
- Structurer les étapes de résolution d'un problème.





Sommaire



- * Présentation
- * Structure de données.
- * Structure algorithmique.
- * Structure de programme.





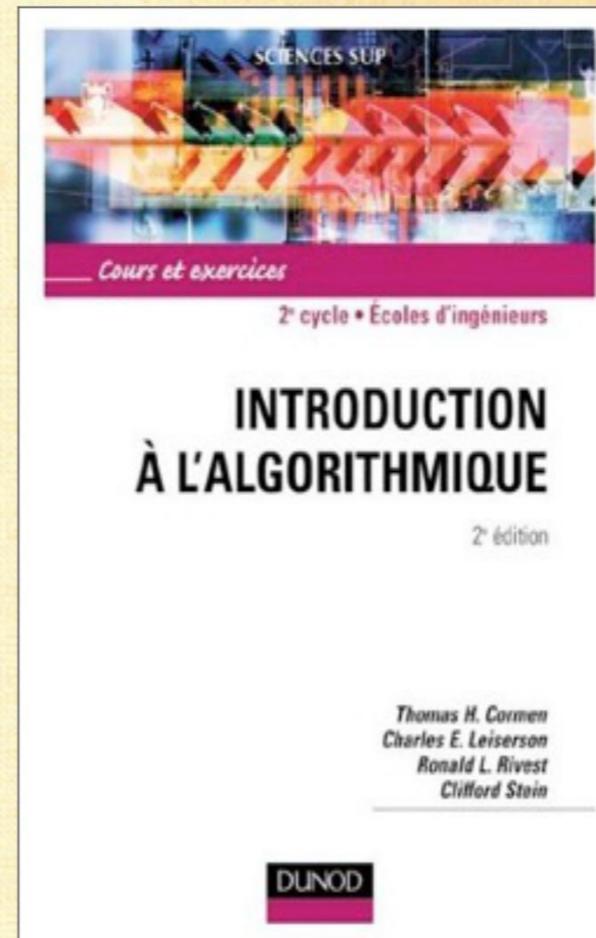
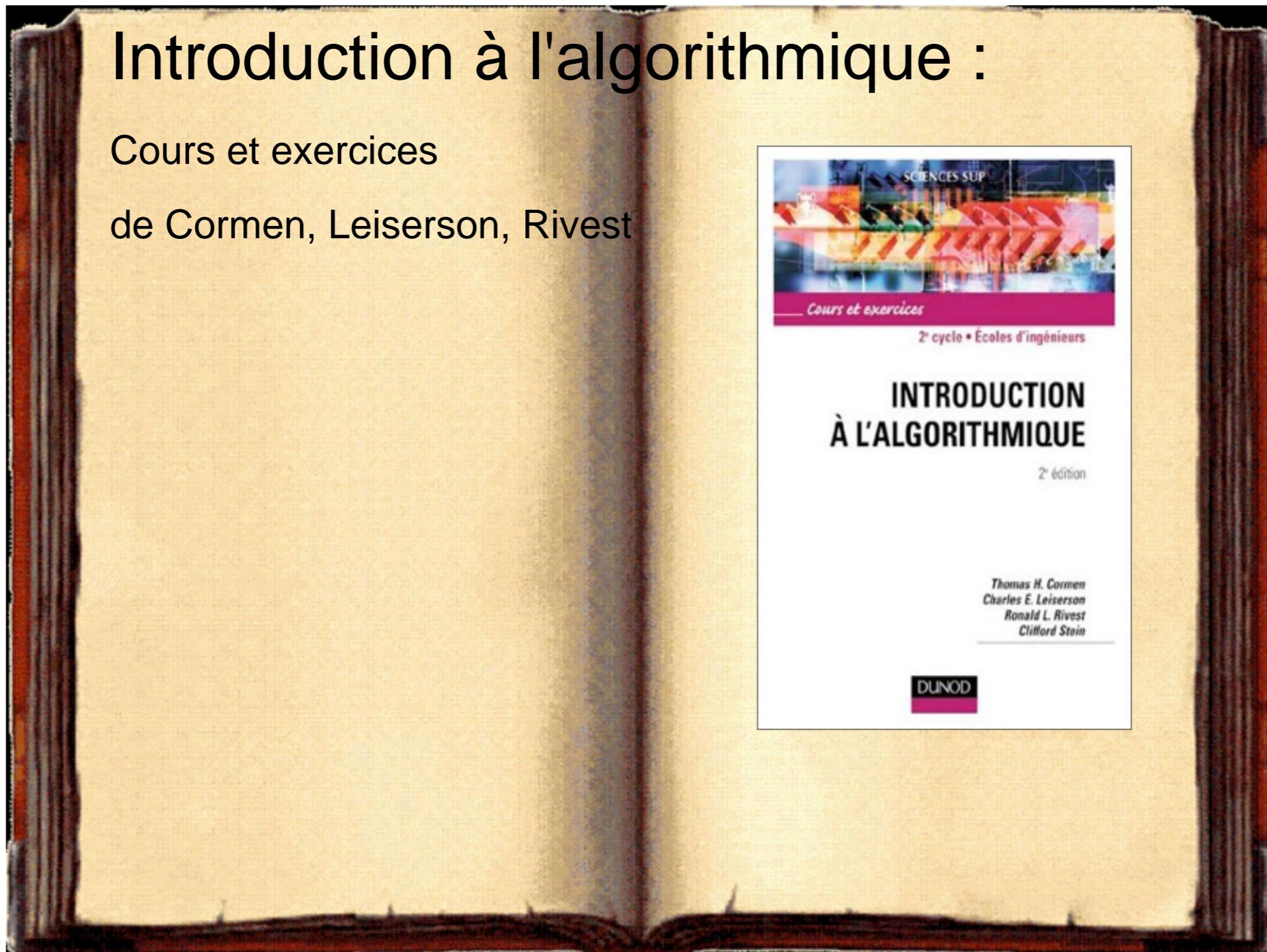
Références Internet

Introduction à l'algorithmique et à la programmation

<http://perso.citi.insa-lyon.fr/afraboul/imsi/algo-imsi-4.pdf>

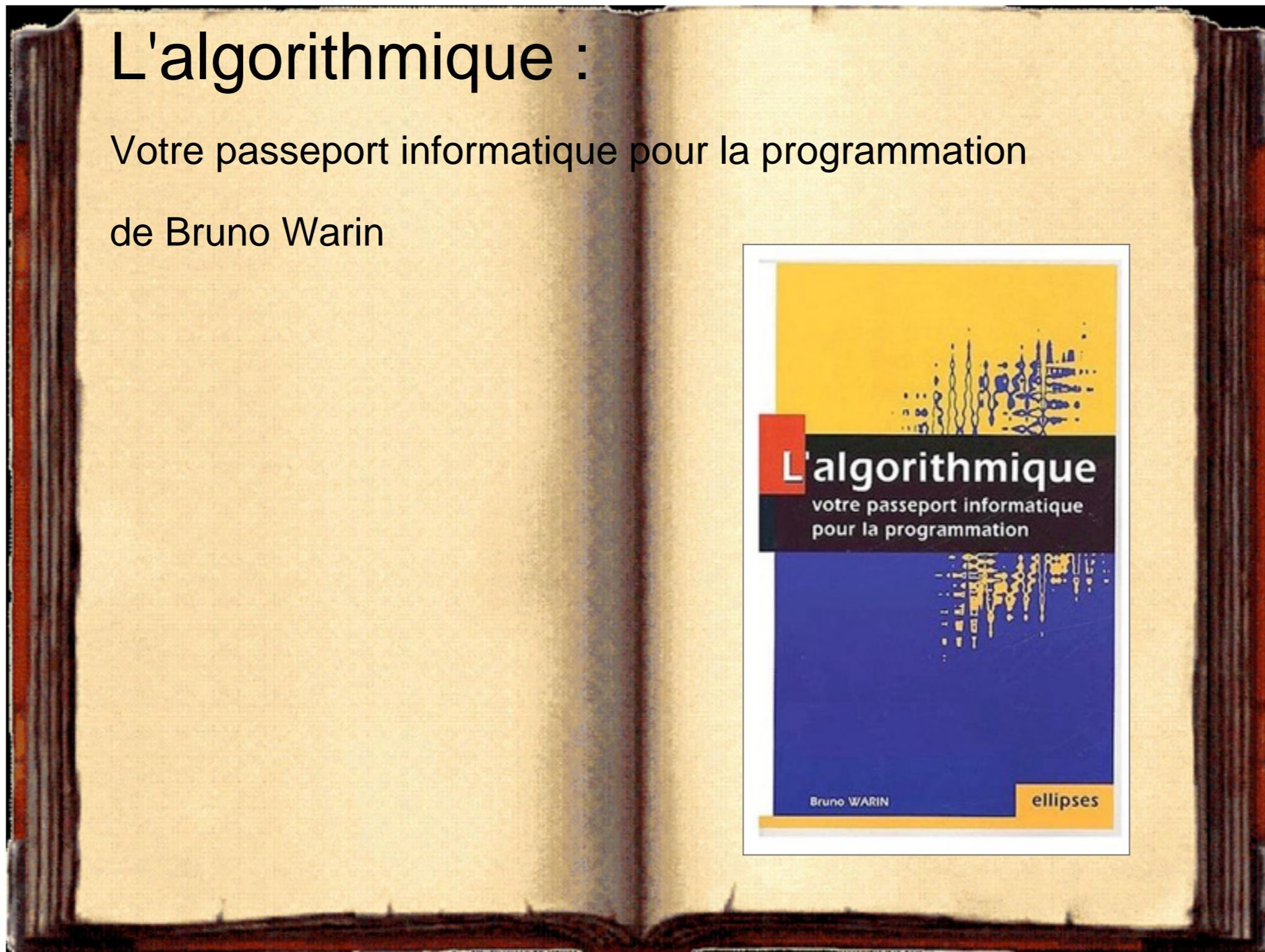


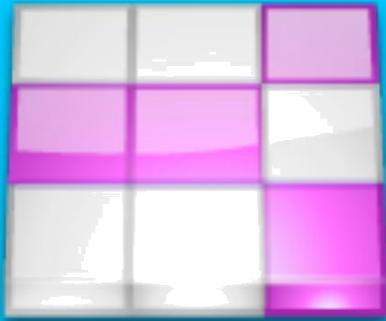
Références Bibliographiques





Références Bibliographiques





Liens pédagogiques



Pré-requis:

Historique et modèle de Von Neumann



Nécessaire pour:

Réaliser tout programme.



Algorithmie

Partie N°2

Structure de données



Algorithmie

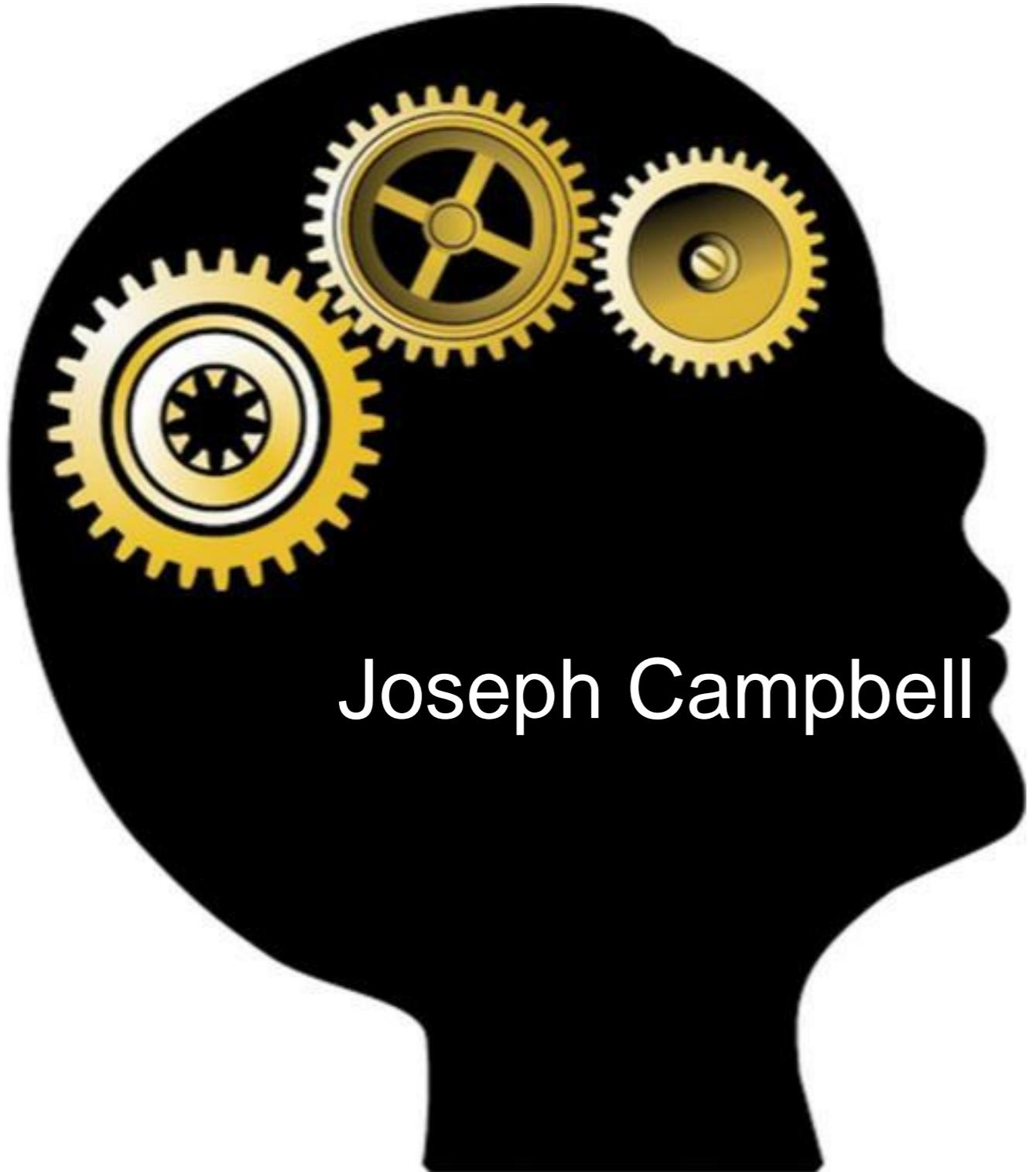
Structure de données

- Présentation
- Les 5 structures de données élémentaires
- Opérateurs et symboles associés
- Opérations élémentaires
- Les structures de données avancées





Citation



Joseph Campbell

«Les ordinateurs sont comme les dieux de l'Ancien Testament : avec beaucoup de règles, et sans pitié.»

Algorithmie > Structure de données

Notion de variables (Rappel)

Les types élémentaires possibles sont:

- ★ Les entiers
- ★ Les réels
- ★ Les caractères
- ★ Les chaînes de caractères
- ★ Les booléens

Algorithmie > Structure de données

Les entiers

ENTIER

nombres

entiers négatifs

positifs

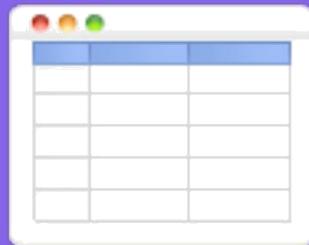
nuls

Exemples :

33 -

13

[pas de limite de taille]



Entier

Opérations possibles

Addition

Soustraction

Multiplication

Division

Exposant

Modulo

Comparaisons

**Opérateurs associés, symboles,
mot clés correspondants**

+

-

DIV (Pour la division entière)

^

MOD (Pour le reste d'une division)

< <= > >= = <>

Algorithmie > Structure de données

Les réels

REEL

 nombres

 entiers

 fractionnés

 négatifs positifs

 nuls

Exemples :

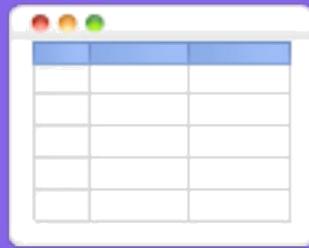
3,26 -10,5

[pas de limite de taille]



Conseil

En écriture algorithmique on utilise
la virgule et non le point pour noter
la partie décimale



Réel

Opérations possibles

Addition
Soustraction
Multiplication
Division
Exposant
Comparaisons

Opérateurs associés, symboles, mot clés correspondants

+

-

*

/

^

< <= > >= = <>

Algorithmie > Structure de données

Les caractères

CARACTÈRE (ou CAR)

caractère unique

Exemples :

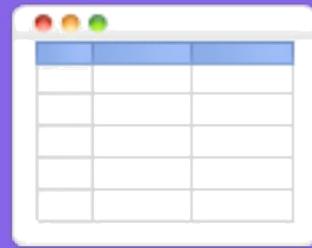
‘F’

‘j’

‘!’

[référence : table des codes ASCII, Unicode]

apostrophe simple



Caractère

Opérations possibles

Comparaisons
Extraction

**Opérateurs associés, symboles,
mot clés correspondants**

< <= > >= = <>

CAR1 ← EXTRACTION (chaine, position)

Algorithmie > Structure de données

Les chaînes

CHAINE

caractère unique

suite de caractères

Exemples :

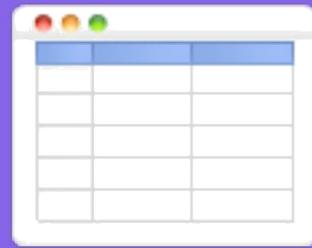
"Bonjour"

"64"

"Super Mario World"

[référence : table des codes ASCII, Unicode]

apostrophe double



Chaîne

Opérations possibles

Comparaisons

Extraction

Longueur

Concaténation

Opérateurs associés, symboles,
mot clés correspondants

< <= > >= = <>

CH1 ← EXTRACTION (CH2, position, longueur)

ent ← LONGUEUR (ch)

ch ← CONCATENER (ch1, ch2)

ou ch ← ch1 & ch2



Conseil



Dans une fonction de chaîne de caractères, un caractère peut être vue comme une chaîne de longueur

1.

Ainsi ,

`chr ← ch2 & '.`

est valide

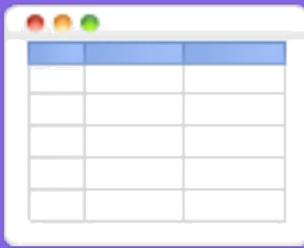
Algorithmie > Structure de données

Les booléens

BOOLEEN

vrai

faux



Booléen

Opérations possibles

Comparaisons

Négation

Conjonction

Disjonction

**Opérateurs associés, symboles,
mot clés correspondants**

= <>

NON

ET

OU

Algorithmie > Structure de données

Autres structure de données

Structures linéaires :

Tableaux

Piles

Files

Listes

Structures non linéaires :

Enregistrements

Graphes

Arbres

Algorithmie > Structure de données

Quelle structure de données ?

"Validez"

ENTIER

FAUX

CHAINE

1024

BOOLEEN

'o'

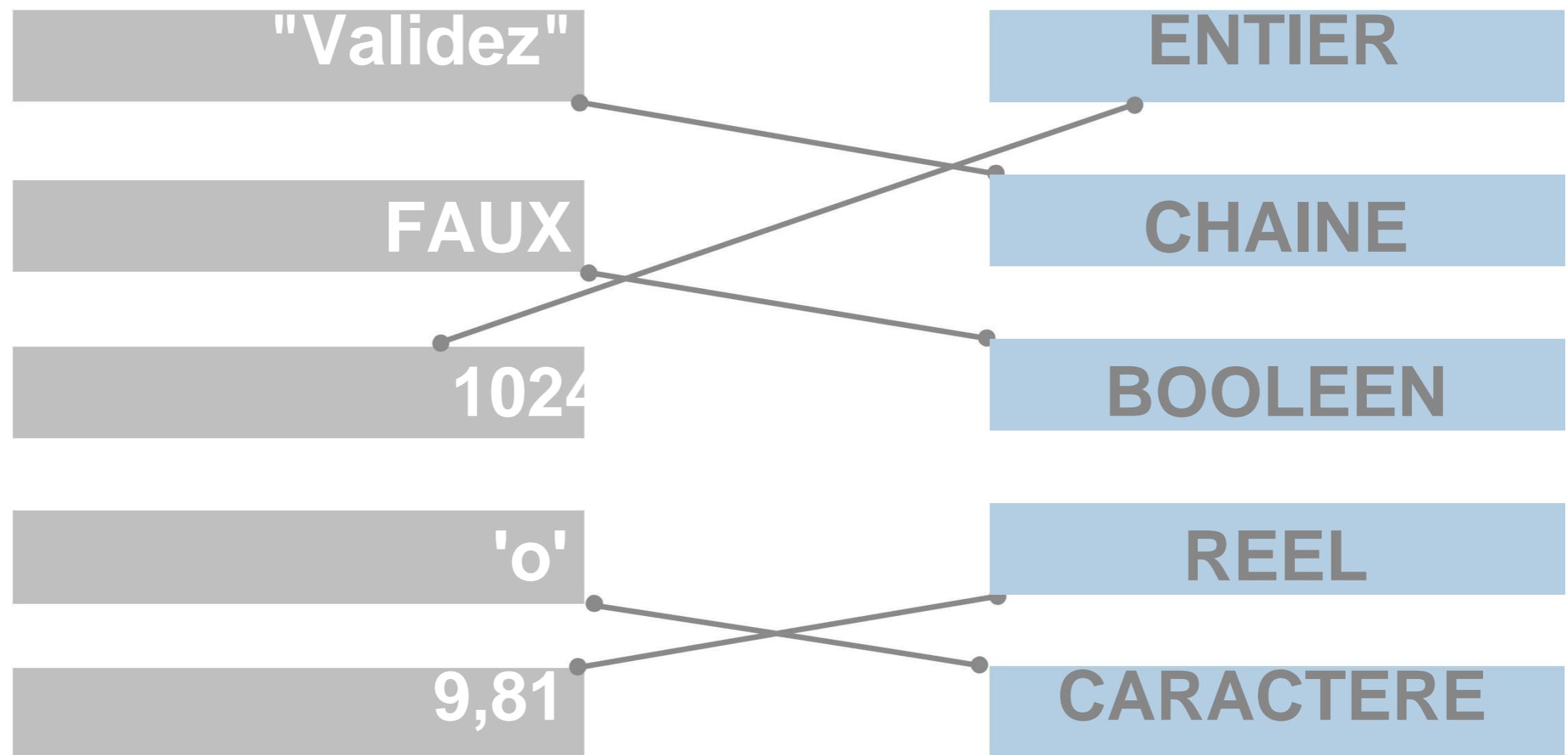
REEL

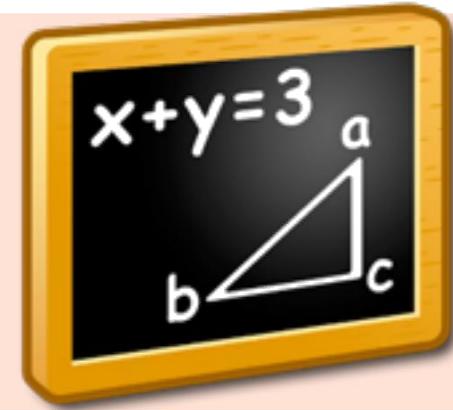
9,81

CARACTERE

Algorithmie > Structure de données

Quelle structure de données ?

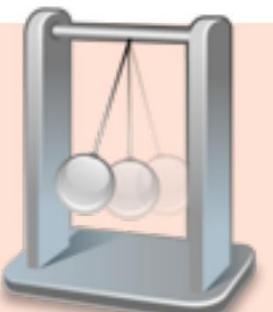




Exercice



Ecrire un algorithme qui demande à un utilisateur son nom et son prénom et qui affiche ses initiales.



5 min



solo



Live



Solution



ALGORITHME : Initiales

//BUT : cet algorithme extrait les initiales d'une personne //ENTREE : Le nom et le prénom saisis par l'utilisateur //SORTIE : les initiales

VAR : nom, prenom, initiales: **CHAINE**

DEBUT

ECRIRE "Veuillez entrer votre
 nom" **LIRE** nom

ECRIRE "Veuillez entrer votre prénom"

LIRE prenom

 initiales \leftarrow **EXTRACTION**(nom,1,1) &

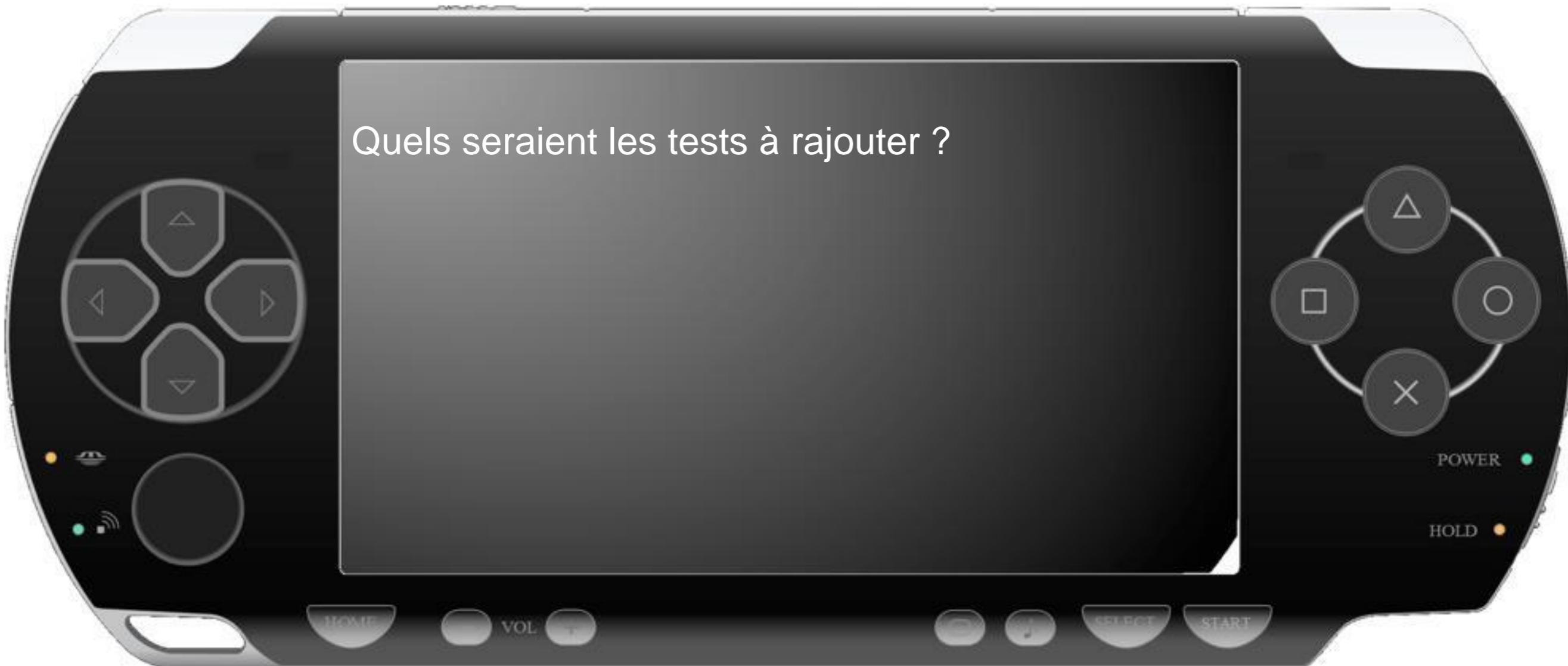
EXTRACTION(prenom,1,1) **ECRIRE** initiales

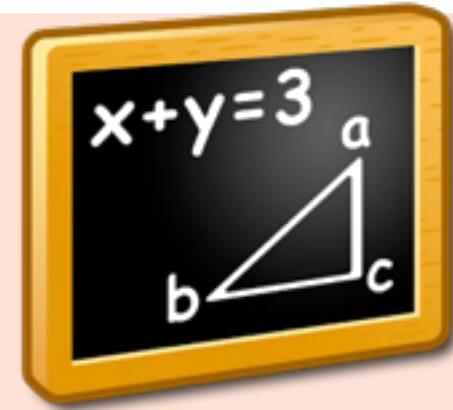
FIN



Question

Quels seraient les tests à rajouter ?



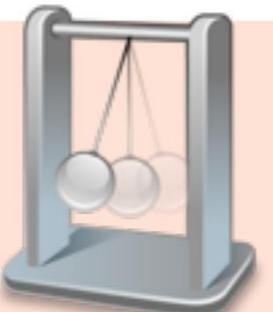


Exercice



Ecrire un algorithme qui demande à un utilisateur un mot de trois lettres et affiche son miroir.

ex: ABC devient CBA



5 min



solo



Live



Solution

ALGORITHME : Mirroir

//BUT : cet algorithme inverse un mot de trois lettres

//ENTREE : Le mot de trois lettres saisi par
l'utilisateur //SORTIE : le mot miroir

VAR : mot, motmirroir: **CHAINE**

DEBUT

ECRIRE "Veuillez entrer votre

mot" LIRE mot

motmirroir \leftarrow EXTRACTION(mot,3,1) & EXTRACTION(mot,2,1) & EXTRACTION(mot,1,1)

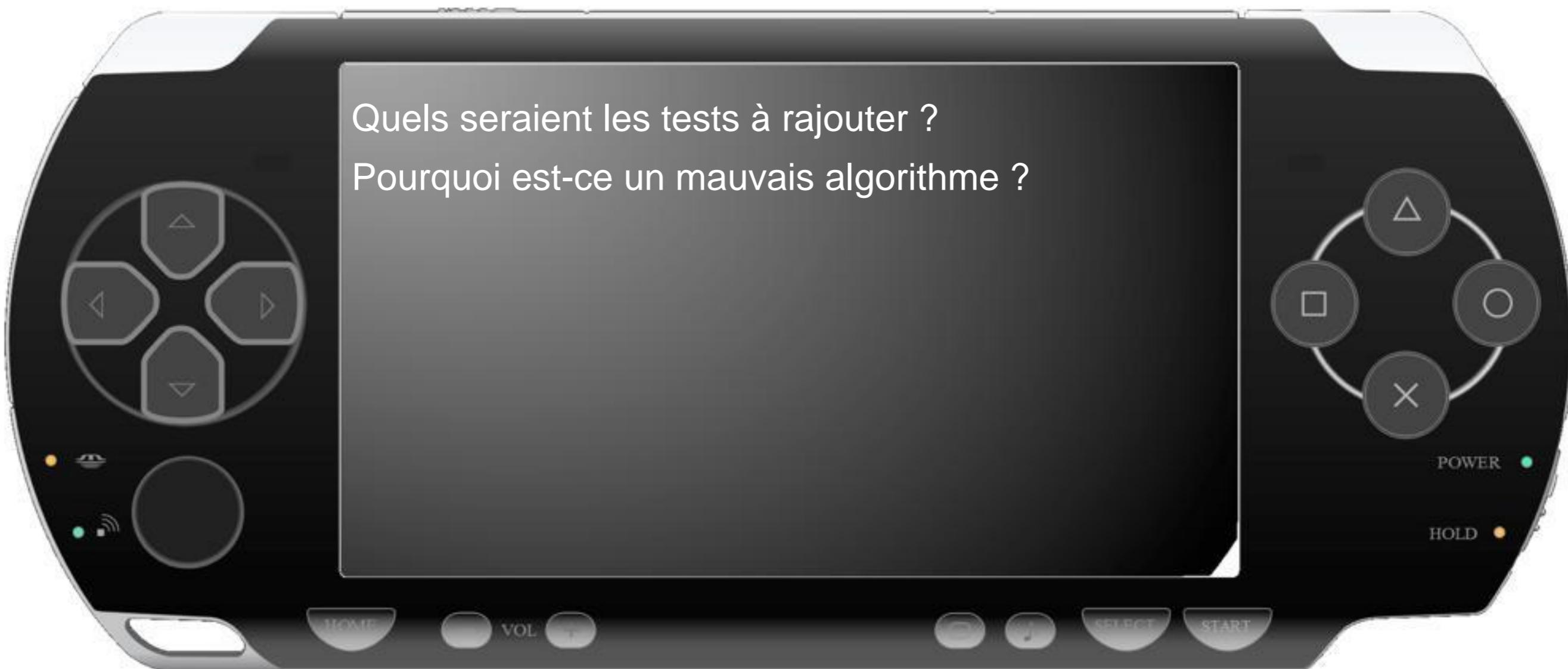
ECRIRE motmirroir

FIN



Question

Quels seraient les tests à rajouter ?
Pourquoi est-ce un mauvais algorithme ?





Solution



ALGORITHME : Mirroir

```
//BUT : cet algorithme inverse un mot de trois lettres
//ENTREE : Le mot de trois lettres saisi par
l'utilisateur //SORTIE : le mot miroir
```

VAR : mot, motmirroir: **CHAINE**

DEBUT

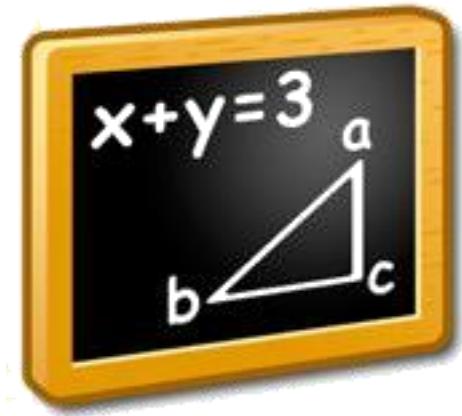
ECRIRE "Veuillez entrer votre
mot" LIRE mot

mot ← EXTRACTION(mot & "___",1,3)

motmirroir ← EXTRACTION(mot,3,1) & EXTRACTION(mot,2,1) & EXTRACTION(mot,1,1)

ECRIRE motmirroir

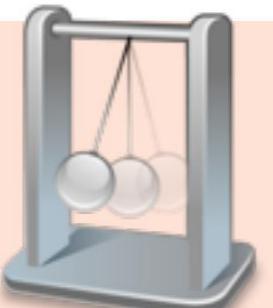
FIN



Ecrire un algorithme qui demande à un utilisateur un mot de cinq lettres et qui affiche si c'est un palindrome.

ex: « RADAR » affiche VRAI

« Mario » affiche FAUX



10 min



solo



Live



Solution

ALGORITHME : Palin

//BUT : cet algorithme vérifie si un mot est un palindrome
//ENTREE : Le mot saisi par l'utilisateur
//SORTIE : Vrai si c'est un palindrome, faux
sinon **VAR** : mot, motmirroir : **CHAINE**

DEBUT

ECRIRE "Veuillez entrer votre
mot" LIRE mot

mot \leftarrow EXTRACTION(mot & "_____",1,5)

motmirroir \leftarrow EXTRACTION(mot,5,1) & EXTRACTION(mot,4,1) & EXTRACTION(mot,3,1)
& EXTRACTION(mot,2,1) & EXTRACTION(mot,1,1)

ECRIRE motmirroir = mot

FIN



Solution

ALGORITHME : Palin

```
//BUT : cet algorithme vérifie si un mot est un
palindrome //ENTREE : Le mot saisi par l'utilisateur
//SORTIE : Vrai si c'est un palindrome, faux
sinon VAR : mot, motmirroir : CHAINE
estpalin: BOOLEEN
```

DEBUT

ECRIRE "Veuillez entrer votre
mot" LIRE mot

mot ← EXTRACTION(mot & "_____",1,5)

estpalin ← EXTRACTION(mot,5,1) = EXTRACTION(mot,1,1) et
EXTRACTION(mot,4,1) = EXTRACTION(mot,2,1)

ECRIRE estpalin

FIN



Solution



ALGORITHME : Palin

//BUT : cet algorithme vérifie si un mot est un palindrome
//ENTREE : Le mot saisi par l'utilisateur
//SORTIE : Vrai si c'est un palindrome, faux sinon

VAR : mot: **CHAINE**

DEBUT

ECRIRE "Veuillez entrer votre
mot" LIRE mot

ECRIRE EXTRACTION(mot & "____",5,1) = EXTRACTION(mot & "_",1,1) et
EXTRACTION(mot & "___",4,1) = EXTRACTION(mot& "__",2,1)

FIN



Algorithmie

Partie N°3

Structure algorithmique



Algorithmie

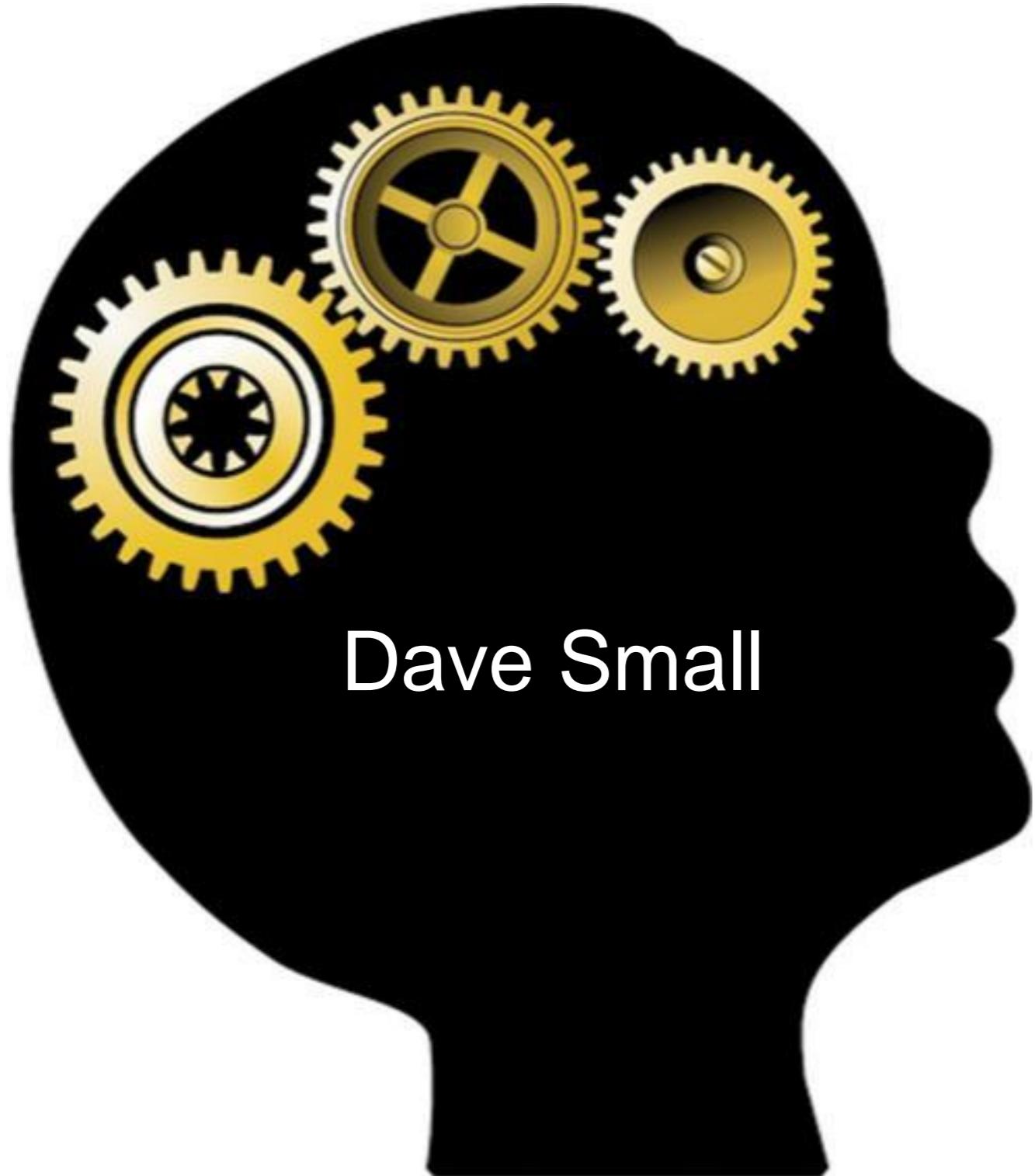
Structure algorithmique

- Présentation
- Rupture de séquence
- Les instructions conditionnelles
- Les instructions répétitives

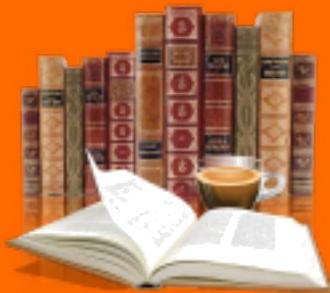




Citation



« Un langage de programmation est une convention pour donner des ordres à un ordinateur. Ce n'est pas censé être obscur, bizarre et plein de pièges subtils. Ca, ce sont les caractéristiques de la magie. »



Définition

Structure Algorithmique:

La structure algorithmique concerne la concaténation et les imbrications de séquences et ruptures de séquences.

Elle forme ainsi une partie d'un programme.

Algorithmie > Structure algorithmique

Catégories

Séquences

Ruptures de séquences :

Rupture conditionnelle (et non répétitive):

SI

CAS

Rupture répétitive :

POUR

TANTQUE

REPETER

Algorithmie > Structure algorithmique

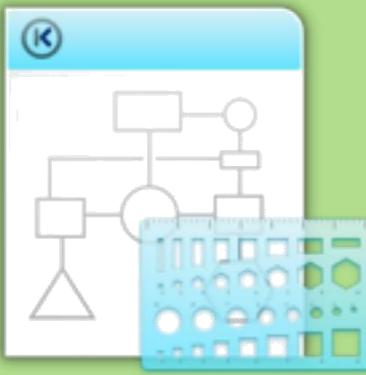
Ruptures conditionnelles

SI..ALORS..FINSI

SI..ALORS..SINON..FINSI

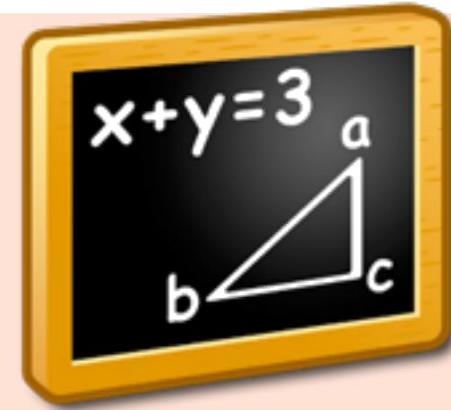
CAS..PARMI..

CAS..PARMI..PARDEFAUT



SI

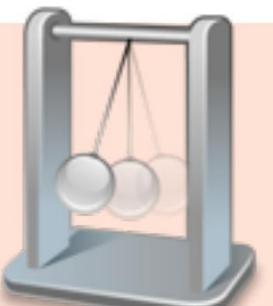
**SI <condition> ALORS
<instructions>
[SINON
<instructions>
]
FINSI**



Exercice



Ecrire un algorithme qui donne le maximum entre deux nombres saisis par l'utilisateur.



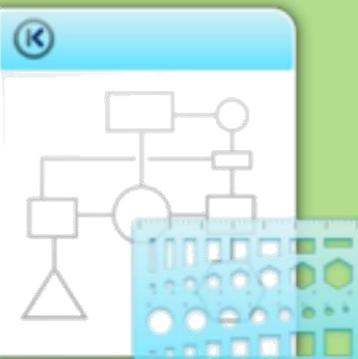
5 min



solo



Live



Algorithme

ALGORITHME: max_deux_nombres

//BUT : cherche la valeur max. parmi 2 valeurs
saisies //ENTREE : deux réels saisis par
l'utilisateur //SORTIE : le maximum des deux valeurs

VAR:

a, b, max :REEL

DEBUT

ECRIRE "Veuillez entrer deux nombres"

LIRE a, b

SI (a >= b)

ALORS

 max ← a

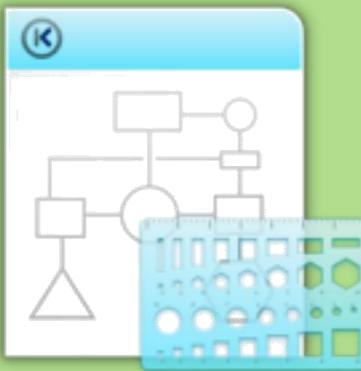
SINON

 max ← b

FINSI

ECRIRE "Le Maximum entre " & a & " et " & b & " est: " & max

FIN



Algorithme

ALGORITHME: max_deux_nombres

//BUT : cherche la valeur max. parmi 2 valeurs
saisies //ENTREE : deux réels saisis par
l'utilisateur //SORTIE : le maximum des deux valeurs

VAR:

a, b, max :REEL

DEBUT

ECRIRE "Veuillez entrer deux nombres"

LIRE a, b

max \leftarrow b

SI (a \geq b)

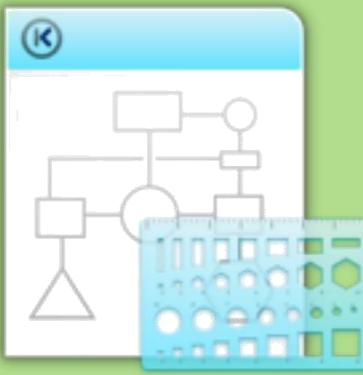
ALORS

max \leftarrow a

FINSI

ECRIRE "Le Maximum entre " & a & " et " & b & " est: " & max

FIN



Algorithme

ALGORITHME: max_deux_nombres

//BUT : cherche la valeur max. parmi 2 valeurs
saisies //ENTREE : deux réels saisis par
l'utilisateur //SORTIE : le maximum des deux valeurs

VAR:

a, b :REEL

DEBUT

ECRIRE "Veuillez entrer deux nombres"

LIRE a, b

SI (a >= b)

ALORS

 ECRIRE "Le Maximum entre " & a & " et " & b & " est: " & a

SINON

 ECRIRE "Le Maximum entre " & a & " et " & b & " est: " & b

FINSI

FIN



Conseil



Le code de l'algorithme
car il a été correctement
Vous devez vous astreindre se lit très bien
«indenté»
à rendre le

code lisible

De même vous devez nommer vos
variables de façon claire et précise

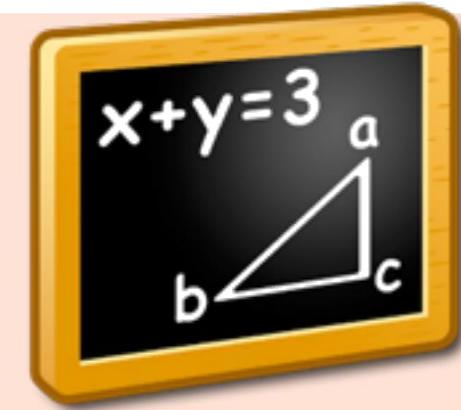
`z ← x * y * const1`

n'évoque

rien

`PrixTTC ← PrixHT * Qte * Taux-TVA`
ne demande aucun effort de

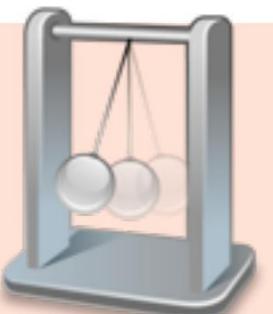
compréhension



Exercice



Ecrire un algorithme qui calcule la valeur absolue d'un entier.



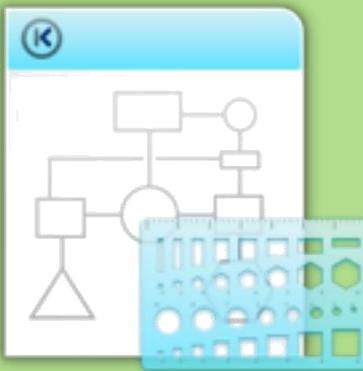
5 min



solo



Live



Algorithme

ALGORITHME: valeur_absolue

//BUT : calcule la val.abs. d'un entier saisi par l'util.
//ENTREE : un entier relatif saisi par l'utilisateur
//SORTIE : la valeur absolue de l'entier saisi

VAR:

x :ENTIER

DEBUT

LIRE x //affectation d'une valeur par l'util. **SI** (x < 0) //si cette valeur est négative

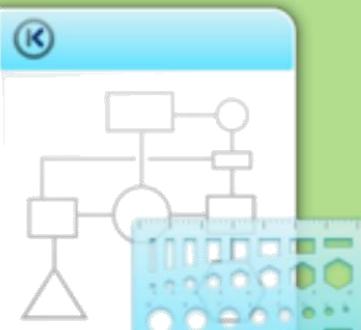
ALORS //alors cette condition est vraie

x \leftarrow x*-1 //et la variable se trouve réaffectée

FINSI

ECRIRE "La val.abs. du nombre saisi vaut : ", x

FIN



Algorithme

ALGORITHME: valeur_absolue

//BUT : calcule la val.abs. d'un entier saisi par l'util.
//ENTREE : un entier relatif saisi par l'utilisateur
//SORTIE : la valeur absolue de l'entier saisi

VAR:

x, abs : **ENTIER**

DEBUT

LIRE x //affectation d'une valeur par l'util. **SI** (x < 0) //si cette valeur est négative

ALORS //alors cette condition est vraie abs ← x*-1

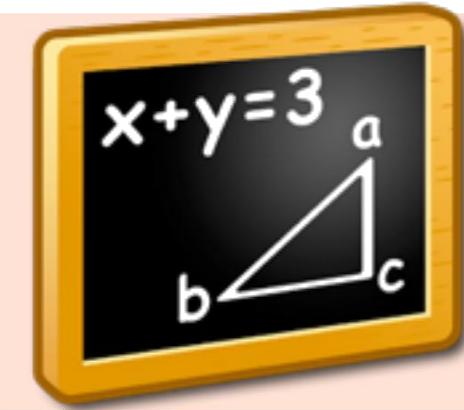
SINON

abs ← x

FINSI

ECRIRE "La val.abs. de " & x & "vaut : " & abs

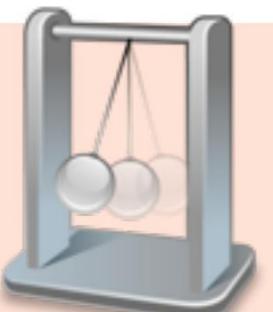
FIN



Exercice



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).



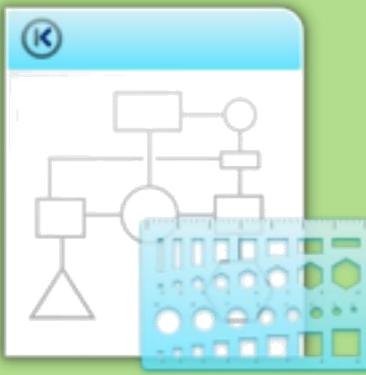
10 min



solo



Live



Algorithme

ALGORITHME: exo-1

VAR:

n : REEL

DEBUT

ECRIRE " Entrez un nombre

" LIRE n

SI (n > 0)

ALORS //alors cette condition est vraie ECRIRE

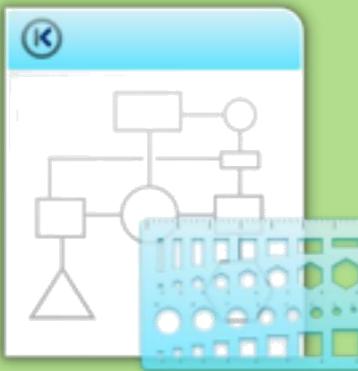
"Ce nombre est positif"

SINON

ECRIRE "Ce nombre est négatif"

FINSI

FIN



Algorithme

ALGORITHME: exo-1

VAR:

n : **REEL**
mot : **CHAINE**

DEBUT

ECRIRE " Entrez un nombre "

LIRE n

SI (n > 0)

ALORS //alors cette condition est vraie

mot \leftarrow "positif"

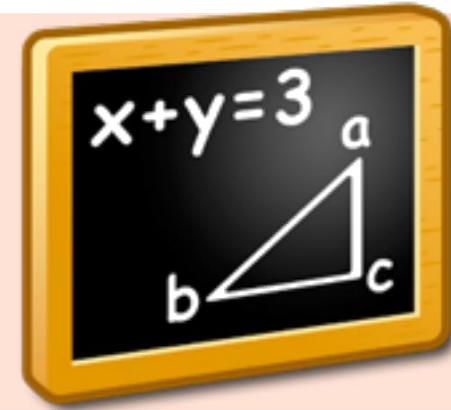
SINON

mot \leftarrow "négatif"

FINSI

ECRIRE "Le nombre " & n & " est " & mot

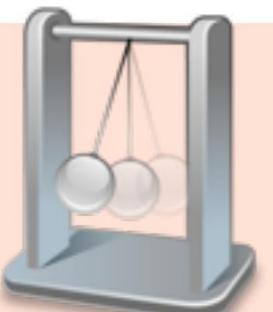
FIN



Exercice



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).



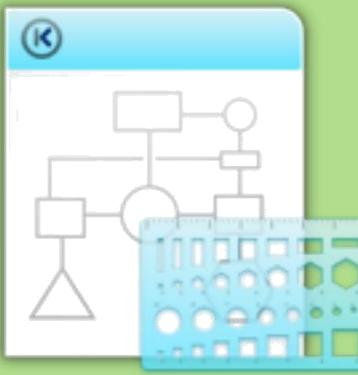
10 min



solo



Live



Algorithme

ALGORITHME: exo-2

VAR:

m, n	:REEL
mot	:CHAINE

DEBUT

ECRIRE " Entrez deux nombres "

LIRE m, n

SI $m * n > 0$

ALORS

mot \leftarrow "positif"

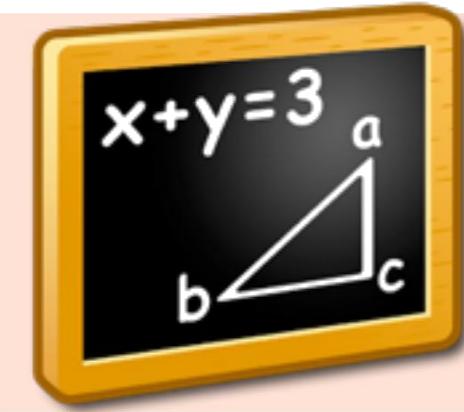
SINON

mot \leftarrow "négatif"

FINSI

ECRIRE "Le produit de " & m & " et " & n & " est " & mot

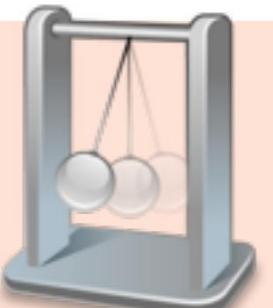
FIN



Exercice



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.



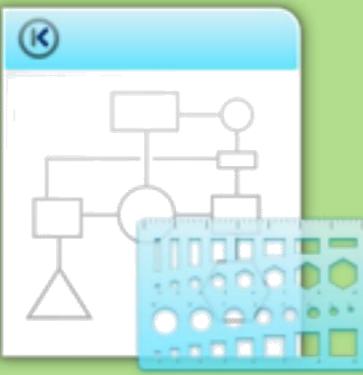
10 min



solo



Live



Algorithme

ALGORITHME: exo-3

VAR:

m, n :ENTIER

DEBUT

ECRIRE " Entrez deux nombres

" LIRE m, n

SI ((m > 0 ET n > 0) OU (m < 0 ET n < 0))

ALORS

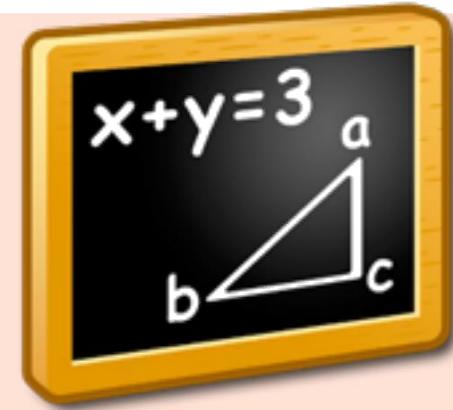
ECRIRE "Le produit est positif"

SINON

ECRIRE "Le produit est négatif"

FINSI

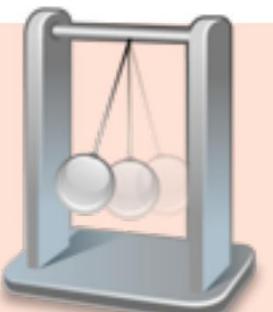
FIN



Exercice



Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.



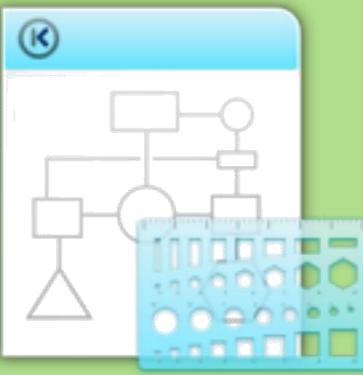
5 min



solo



Live



Algorithme

ALGORITHME: exo-4

VAR:

m1, m2, m3 : CHAINE

DEBUT

ECRIRE " Entrez trois noms

" LIRE m1, m2, m3

SI ((m1<=m2) et (m2<=m3))

ALORS

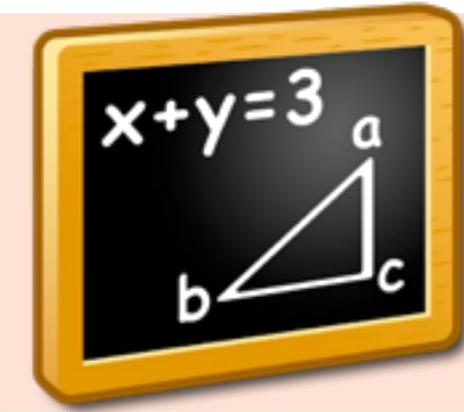
ECRIRE "Ces noms sont classés"

SINON

ECRIRE "Ces noms ne sont pas classés"

FINSI

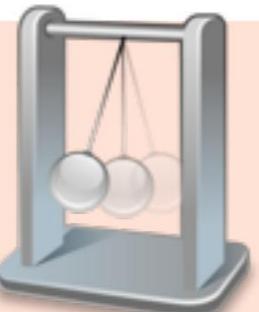
FIN



Exercice



Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif, négatif ou nul.



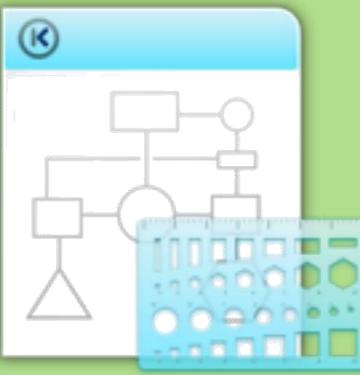
5 min



solo



Live



Algorithme

ALGORITHME: exo-1

VAR:

n : **REEL**

DEBUT

ECRIRE " Entrez un nombre

" LIRE n

SI (n > 0) **ALORS**

 ECRIRE "Ce nombre est positif"

SINON

SI (n < 0) **ALORS**

 ECRIRE "Ce nombre est négatif"

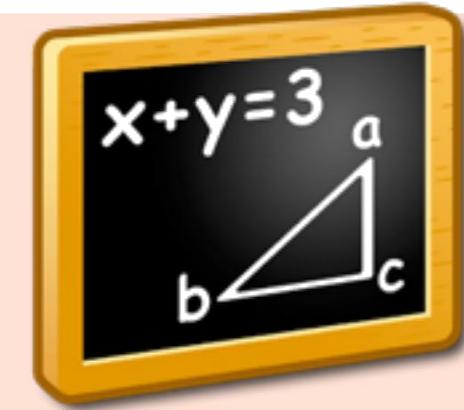
SINON

 ECRIRE "Ce nombre est nul "

FINSI

FINSI

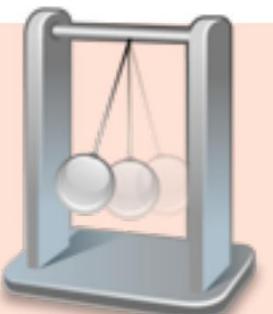
FIN



Exercice



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 !



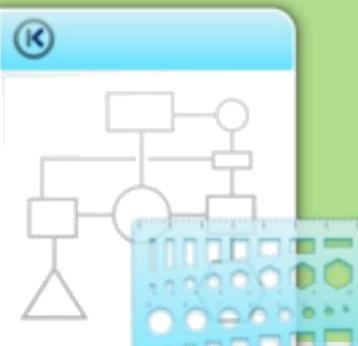
5 min



solo



Live



Algorithme

ALGORITHME: exo-3

VAR:

m, n : ENTIER

DEBUT

ECRIRE " Entrez deux nombres

" LIRE m, n

SI ((m > 0 ET n > 0) OU (m < 0 ET n < 0))

ALORS ECRIRE "Le produit est positif"

SINON

SI (m = 0 OU n = 0) **ALORS** ECRIRE

 "Le produit est nul »

 SINON

 ECRIRE "Le produit est négatif"

FINSI

FINSI

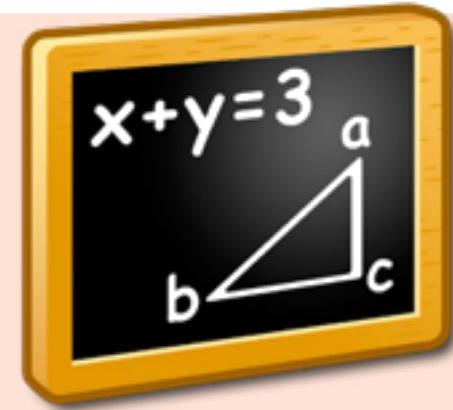
FIN



Conseil



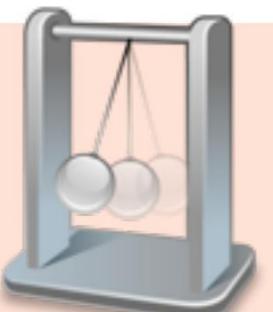
On a ici des structures IMBRIQUEES
L'art de la programmation est de
combiner et d'imbriquer intelligemment
ces structures



Exercice



Ecrire un algorithme qui demande le numéro d'un mois de l'année et affiche le nom du mois.



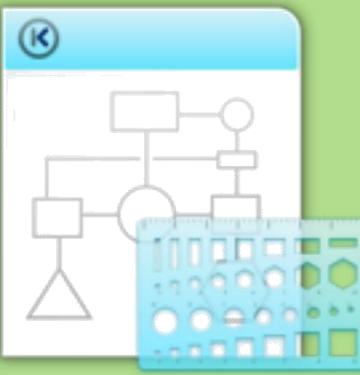
15 min



solo



Live



Algorithme

ALGORITHME: mois1

VAR:

mois : **ENTIER**
mois_chaine : **CHAINE**

DEBUT

LIRE mois
SI (mois = 1)

ALORS

mois_chaine \leftarrow "janvier "

FINSI

SI (mois = 2)

ALORS

mois_chaine \leftarrow "février "

FINSI

...

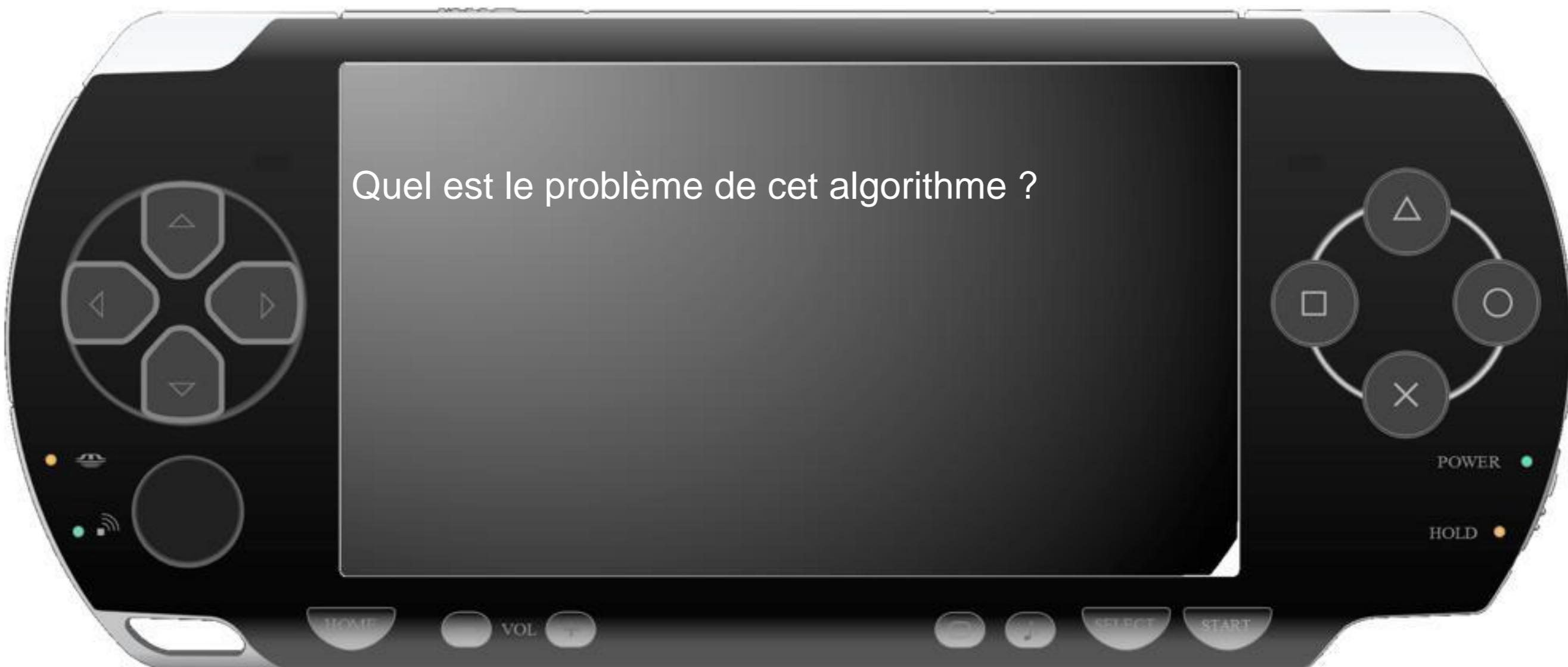
ECRIRE "Le mois est : ", mois_chaine

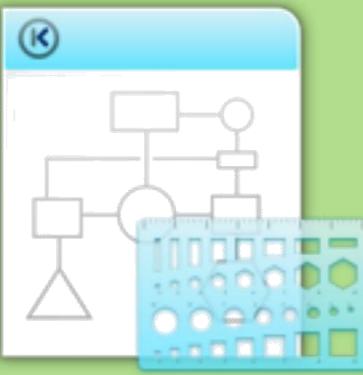
FIN



Question

Quel est le problème de cet algorithme ?





Algorithme

ALGORITHME: mois2

VAR:

mois :ENTIER
mois_chaine :CHAINE

DEBUT

LIRE mois

SI (mois = 1)

ALORS

mois_chaine \leftarrow "janvier "

FINSI

SI (mois = 2)

ALORS

mois_chaine \leftarrow "février "

FINSI

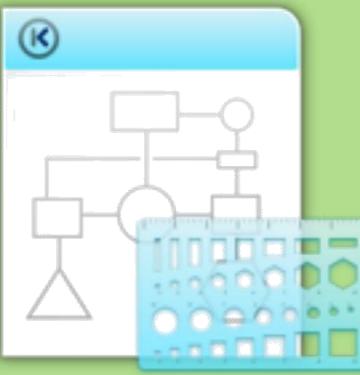
...

SI mois \geq 1 et mois \leq 12 **ALORS**

ECRIRE "Le mois est : ", mois_chaine

FIN SI

FIN



Algorithme

ALGORITHME: mois3

VAR:

mois :ENTIER
mois_chaine :CHAINE

DEBUT

LIRE mois

mois_chaine \leftarrow ""

SI (mois = 1)

ALORS

mois_chaine \leftarrow "janvier "

FINSI

SI (mois = 2)

ALORS

mois_chaine \leftarrow "février "

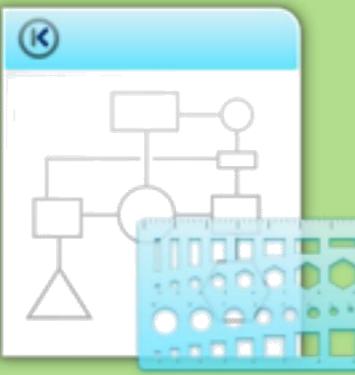
FINSI

...

SI mois_chaine \neq "" **ALORS**

ECRIRE "Le mois est : ", mois_chaine **FINSI**

FIN



Algorithme

ALGORITHME : mois4

VAR:

mois	:ENTIER
mois_chaine	:CHAINE

DEBUT

```
LIRE mois
SI ( mois = 1 ) ALORS
    mois_chaine ← "janvier "
SINON SI ( mois = 2 )
    ALORS
        mois_chaine ← "février"
    " SINON SI ( mois = 3 )
        ALORS
            mois_chaine ← "mars "
        SINON
            ...
FINSI FINSI FINSI
FINSI
    ECRIRE "Le mois est : ", mois_chaine
FIN
```

