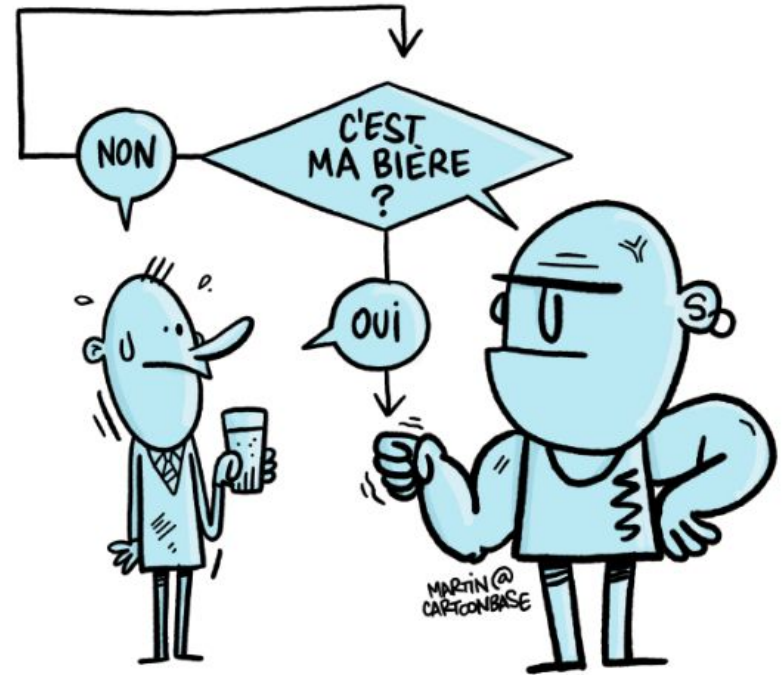


Algorithmie

Session 1
Histoire
Boucle



UN ALGORITHME NE RÉSOUD PAS TOUT

Algorithme

- Mot dérivé du nom du mathématicien “al_Khwarizmi” qui a vécu au 9ème siècle, était membre de l’académie des sciences à Bagdad .
- Un algorithme est une séquence d’instructions qui décrit comment résoudre un problème particulier
- Un algorithme prend des **données en entrée**, exprime **un traitement** particulier et fournit des **données en sortie**.
- Un Programme est série d’instructions pouvant s’exécuter en séquence, ou en parallèle qui implémente un ou plusieurs algorithmes

Algorithme

- **Savoir expliquer** comment faire un travail sans la moindre ambiguïté
- **Langage simple**
- Suite finie d'actions à entreprendre en respectant une chronologie imposée

Un algorithme ne dépend pas

- du langage dans lequel il est implanté.
- ni de la machine qui exécutera le programme correspondant.

Algorithme :: Exemple simple

```
ALGO EleveAuCarre
{cet algorithme calcule le carré du nombre que lui fournit l'utilisateur}

VARIABLES (unNombre, sonCarre: entiers)

DEBUT
    AFFICHER("Quel nombre voulez-vous élever au carré ?")
    SAISIR(unNombre)
    sonCarre <- unNombre * unNombre
    AFFICHER("Le carré de ", unNombre, " c'est ", sonCarre)
FIN
```

Algorithme :: Exemple simple

ALGO CalculTTC

{cet algorithme calcule le prix TTC à partir d'un prix HT fournit l'utilisateur}

VARIABLES (prixHT : flottant)

CONSTANTE tva: 21

DEBUT

C'est a vous

FIN

Algorithme :: Exemple simple (corrigé)

ALGO CalculTTC

{cet algorithme retourne le calcul d'un prix TTC à partir d'un prix HT fournit l'utilisateur}

VARIABLES (prixHT, prixTTC : flottant)

CONSTANTE tva: 21

DEBUT

AFFICHER("Quel prix HT voulez vous le prix TTC ?")

SAISIR(prixHT)

$\text{prixTTC} \leftarrow \text{prixHT} * (1 + \text{tva} / 100)$

AFFICHER("Le prix TTC est : ", prixTTC)

FIN

Algorithme :: Ça fait quoi ? => Un bug

Algorithme CaEchange

{Cet algorithme échange les 2 valeurs fournis en paramètre)

VARIABLES valA, valB: réels

DEBUT

AFFICHER ("Donnez-moi deux valeurs :")

SAISIR (valA, valB)

AFFICHER ("Vous m'avez donné ", valA, " et ", valB)

valA ← valB

valB ← valA

AFFICHER ("Maintenant , mes données sont : ", valA, " et ", valB)

FIN

Algorithme :: c'est corrigé :) (merci les pots de confiture)

Algorithme CaEchange

{Cet algorithme échange les 2 valeurs fournis en paramètre)

VARIABLES valA, valB, valC: réels

DEBUT

AFFICHER ("Donnez-moi deux valeurs :")

SAISIR (valA, valB)

AFFICHER ("Vous m'avez donné ", valA, " et ", valB)

valC ← valB

valA ← valB

valB ← valC

AFFICHER ("Maintenant , mes données sont : ", valA, " et ", valB)

FIN

Algorithme :: Boucle Conditionnelle

Algorithme SimpleOuDouble

{ Cet algorithme saisit une valeur entière et affiche son double si cette donnée est inférieure à un seuil donné. }

CONSTANTE (SEUIL : entier) \leftarrow 10

VARIABLE val : entier

DEBUT

AFFICHER("Donnez-moi un entier : ") { saisie de la valeur entière}

SAISIR(val)

SI val < SEUIL { comparaison avec le seuil} **ALORS**

AFFICHER ("Voici son double :", val \times 2)

SINON

AFFICHER ("Voici la valeur inchangée :", val)

FINSI

FIN

Algorithme :: Boucle Conditionnelle Imbriquée

Algorithme Mention

{ Cet algorithme saisit une valeur entière et affiche son double si cette donnée est inférieure à un seuil donné. }

VARIABLE note : numérique

DEBUT

AFFICHER("Donnez-moi une note : ") { saisie de la valeur numérique}

SAISIR(note)

SI note >= 12 ALORS

AFFICHER ("Reçu avec mention AB")

SINON

SI note >= 10 ALORS

AFFICHER ("Reçu mention Passable")

SINON

AFFICHER ("Non Reçu")

FINSI

FINSI

FIN

Algorithme :: Boucle SUIVANT CAS (SELON)

SUIVANTCAS abréviation

"M" : AFFICHER (" Monsieur ")

"Mme" :AFFICHER (" Madame ")

"Mlle" : AFFICHER (" Mademoiselle ")

AUTRE : AFFICHER (" Monsieur, Madame ")

FINSUIVANTCAS

Algorithme :: Boucle SUIVANT CAS (SELON) ou SI ?

SUIVANTCAS abréviation

"M" : AFFICHER (" Monsieur ")

"Mme" :AFFICHER (" Madame ")

"Mlle" : AFFICHER (" Mademoiselle ")

AUTRE : AFFICHER (" Monsieur, Madame ")

FINSUIVANTCAS

{Équivalent avec instruction Conditionnelle }

SI abréviation = "M " **ALORS**

AFFICHER("Monsieur")

SINON

SI abréviation = « Mlle » **ALORS**

AFFICHER("Mademoiselle")

SINON

SI abréviation = "Mme" **ALORS**

AFFICHER("Madame")

SINON

AFFICHER("Monsieur, Madame ")

FINSI

FINSI

FINSI

Algorithme :: Boucle SUIVANT CAS (SELON) ou SI ?

SUIVANTCAS abréviation

"M" : AFFICHER (" Monsieur ")

"Mme" : AFFICHER (" Madame ")

"Mlle" : AFFICHER (" Mademoiselle ")

AUTRE : AFFICHER (" Monsieur, Madame ")

FINSUIVANTCAS

SI abréviation = "M " **ALORS**

AFFICHER("Monsieur")

FINSI

SI abréviation = « Mlle » **ALORS**

AFFICHER("Mademoiselle")

FINSI

SI abréviation = "Mme" **ALORS**

AFFICHER("Madame")

FINSI

SI abréviation <> "M "

ET abréviation <> « Mlle »

ET abréviation <> « Mme » **ALORS**

AFFICHER("Monsieur, Madame ")

FINSI

Algorithme :: Boucle POUR

Répéter une suite d'instructions un certain nombre de fois

```
POUR <var> <- valInitial A valFin [PAR <pas>]  
FAIRE  
    traitement {suite d'instructions}  
FINPOUR
```

valInitial :: Valeur Initiale

valFin :: Valeur Finale

Pas : Valeur à ajouter à <var> à chaque passage dans la boucle

Algorithme :: Boucle TANTQUE ... FAIRE

Répéter un traitement a nombre d'itération inconnu

```
| amorçage  
| TANTQUE <expression logique vraie> FAIRE  
|     traitement {suite d'instructions}  
|     relance  
| FINTANTQUE
```

amorçage :: initialisation de la (des) variable(s) de condition

traitement :: suite d'instructions à exécuter si condition vraie

relance : ré-affectation de la (des) variable(s) de condition

Algorithme :: Boucle POUR ou TANTQUE ... FAIRE ?

```
POUR cpt <- 0 A 10 PAR 1 FAIRE  
  AFFICHER ("Donnez une valeur :")  
  SAISIR(valeur)  
  totalValeurs <- totalValeurs + valeur  
FINPOUR
```

```
cpt <- 0  
TANTQUE cpt < 10 FAIRE  
  AFFICHER ("Donnez une valeur :")  
  SAISIR(valeur)  
  totalValeurs <- totalValeurs + valeur  
  cpt <- cpt + 1  
FINTANTQUE
```


Algorithme :: Boucle POUR ou TANTQUE ... FAIRE ?

- Nombre d'itération connu à l'avance : POUR
- Boucle s'arrête sur événement particulier : TANT QUE

Algorithme :: Boucle FAIRE ... JUSQU'A ?

FAIRE

AFFICHER ("Donnez une valeur positive non nulle :")

SAISIR(valeur)

JUSQU'A valeur ≥ 0

AFFICHER ("La valeur positive non nulle que vous avez saisie est ", valeur)

Algorithme: Exercice DM ?

- Algorithme de l'ascenseur
 - Immeuble de 4 niveaux (RDC + 3 étages)
 - Les paliers ont 1 bouton d'appel (le même bouton)
 - Utilisation de fonction(Algorithme a fournir en bonus)
 - ALLER_A(numeroEtagCible)
 - Utilisation des fonctions ci-dessous (si vous en avez besoin)
 - MONTER_DE_1_ETAGE() // pas de paramètre
 - DESCENDRE_DE_1_ETAGE() // pas de paramètre
 - OUVERTURE_PORTE() // pas de paramètre
 - FERMETURE_PORTE() // pas de paramètre
 - ATTENDRE(nombreSeconde) // paramètre nombre de seconde avant de faire la poursuite du programme
 - POSITION_ASCENSEUR() // pas de paramètre, mais retourne l'étage actuel de l'ascenseur
 - ETAT_PORTE() // pas de paramètre, mais retourne l'état ouvert ou fermé

Proposer un algorithme, essayer d'identifier tous les cas au limites

Attention l'exercice est plus compliqué qu'il n'y paraît