



COLLEGE
DE PARIS
SUPÉRIEUR

DAKARTECH

ALGORITHMIQUE & PROGRAMMATIONS(C/C++)

Dr Pape Abdoulaye BARRO

Enseignant – Chercheur

Spécialiste en Télémétrie & Systèmes Intelligents

STRUCTURES CONDITIONNELLES ET ITÉRATIVES

STRUCTURES ITÉRATIVES

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

- ✖ Une itération consiste en la répétition d'un blocs d'instructions jusqu'à ce qu'une certaine condition soit vérifiée.
- ✖ Il existe 2 sortes d'itérations :
 - ❖ Le nombre de répétitions est connu dès le départ
Exemple : Calcul de $1+2+3+\dots+n$
 - ❖ Le nombre de répétitions est méconnu
Exemple : Recherche de PGCD de deux entiers

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Considérons le problème suivant:

On se propose d'afficher tous les entiers naturels strictement plus petits que 100. Il est évident que, même s'il est possible de résoudre ce problème en écrivant toutes les instructions comme suit :

Ecrire("1")

Ecrire("2")

..

Ecrire("99")

Cette solution reste non désirée, d'où la nécessité de trouver une alternative. C'est là que la notion de boucle va prendre toute son importance.

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Pour i allant de MIN à MAX par pas de PAS faire
 {instructions}
FinPour

- ✧ **Remarque:** Lorsque le pas d'itération vaut 1, la syntaxe peut être simplifiée comme suit:

Pour i allant de MIN à MAX faire
 {instructions}
FinPour

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Exemple:

Algorithme Factorielle

{ Cet algorithme permet de calculer la factorielle d'un entier naturel }

Variable n, i, resultat : entier

Début

Ecrire("Entrez un entier naturel")

Lire(n)

resultat \leftarrow 1;

Pour i allant de 2 à n **faire**

 resultat \leftarrow resultat *i

Fin pour

Ecrire(n,"! =", resultat)

Fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

× Exemple:

Écrire un programme qui permet de calculer la somme $S=1+2+3+4+....+ N$.
où N saisi au clavier par l'utilisateur.

× Solution:

Algorithme Somme_des_N_entiers

Variables i,S,N: entiers

Debut

$S \leftarrow 0$

Ecrire("Donner un entier")

Lire (N)

Pour i allant de 1 à N faire

$S \leftarrow S + i$

FinPour

Ecrire("La somme est:" , S)

Fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Faire

{instructions}

TantQue (condition_de_reprise)

- ✖ Cette boucle peut s'écrire alternativement comme suit:

Répéter

{instructions}

Jusqu'à(condition_de_sortie)

- ✖ **Remarque** : Préférez la première syntaxe, elle est plus proche de l'implémentation fournie par la plupart des langages de programmation.

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Exemple: avec Faire ... TantQue

Algorithme Factorielle

{ Cet algorithme permet de calculer la factorielle d'un entier naturel }

Variable n, i, result : entier

Debut

Ecrire('Entrez un entier naturel')

Lire(n)

 result \leftarrow 1;

 i \leftarrow 1;

Faire

 result \leftarrow result * i

 i \leftarrow i + 1

TantQue(i \leq n)

Ecrire(n, " ! = ", result)

Fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Exemple: Avec Répéter ... Jusqu'à

Algorithme Factorielle

{ Cet algorithme permet de calculer la factorielle d'un entier naturel }

Variable n, i, result : entier

Debut

Ecrire('Entrez un entier naturel')

Lire(n)

result \leftarrow 1;

i \leftarrow 1;

Répéter

result \leftarrow result * i

i \leftarrow i + 1;

Jusqu'à (i > n)

Ecrire(n, " ! = ", result)

Fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

✖ Exemple:

Écrire un algorithme qui affiche la table de multiplication de 8.

✖ Solution:

Algorithme Table_Multiplication_de_8

Variables i:entier

Debut

$i \leftarrow 0$

Faire

Ecrire("8*",i,"=",i*8)

$i \leftarrow i+1$

TantQue($i \leq 10$)

Fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

```
TantQue (condition_d_entree) faire  
    {instructions}  
FinTantQue
```

- ✖ **Remarque** : La condition d'entrée doit être définie au préalable sinon, en implémentant votre algorithme, vous risquez d'avoir des comportements étranges.

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

Exemple:

Algorithme Factorielle

{ Cet algorithme permet de calculer la factorielle d'un entier naturel }

Variable n, i, result : entier

Debut

Ecrire("Entrez un entier naturel")

Lire(n)

result \leftarrow 1;

i \leftarrow 1;

TantQue(i \leq n) **faire**

 result \leftarrow result * i

 i \leftarrow i + 1

FinTantQue(i \leq n)

Ecrire(n, " ! = ", result)

Fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

✖ **Exemple:**

Écrire un programme permettant de calculer la somme $S=1+2+3+\dots+N$, où N saisi par l'utilisateur.

✖ **Solution:**

Algorithme Somme_de_1_jusqu'au_N

Variables i, S, N : entiers

Debut

$i \leftarrow 1$

$S \leftarrow 0$

Ecrire("Donner un entier:")

Lire (N)

TantQue ($i \leq N$) **faire**

$S \leftarrow S + i$

$i \leftarrow i + 1$

FinTantQue

Ecrire("La somme de 1 à N est:", S)

fin

RAPPEL ALGORITHMIQUE

STRUCTURES ITÉRATIVES

QUELLE BOUCLE CHOISIR ?

Chaque boucle a un contexte dans lequel son utilisation est plus adéquate bien qu'il soit possible d'utiliser l'une comme l'autre dans certains contextes.

- × **Pour** : Lorsque le nombre d'itérations est connu
- × **Faire ... tantQue** : Lorsque nous sommes certains d'exécuter le bloc d'instruction au moins une fois.
- × **TantQue ... faire** : Lors que le bloc d'instruction peut ne pas du tout être exécuté.

RAPPEL ALGORITHMIQUE

CAS PRATIQUES N°3

✖ Application 16 :

Écrire un algorithme permettant de calculer la somme des nombres impairs de 1 à n .

✖ Application 17:

Écrire un algorithme permettant de calculer la moyenne des n premiers entiers. n étant strictement positif.

✖ Application 18:

Écrire un algorithme permettant de lire 20 nombres au clavier et d'afficher le carré des nombres pairs uniquement. Attention, on ne mémorisera pas les 20 valeurs saisies.

✖ Application 19:

Écrire un algorithme qui demande un nombre à l'utilisateur, puis vérifie et affiche que les nombres paires.

Le programme s'arrête lorsque l'utilisateur donne -1.

✖ Application 20:

Écrire l'algorithme permettant de lire puis d'afficher une valeur comprise entre 1 et 31; on recommencera la saisie jusqu'à ce que la valeur soit bien dans les bornes imposées.

Affaires à suivre

