



Filière Génie électrique

Module 5-E2: Informatique

Algorithmes & Langage C

Doctorant: Hamza HAMOUT

Année Universitaire 2019/2020

Avant de commencer

Pourquoi L'informatique ?

Algorithme: Introduction

- **Qu'est-ce qu'un algorithme ?**

Un algorithme est une suite logique d'instructions permettant de résoudre un problème (ou de répondre à un besoin).

Qu'est ce que veut dire « écrire un algorithme »

- Analyser et comprendre le problème : étude des données fournies et des résultats attendus.

- Résoudre le problème :

C'est trouver les structures de données adaptées ainsi que l'enchaînement des actions à réaliser pour passer des données aux résultats.



Énoncé d'un problème

Analyse, compréhension

Résolution

Pseudo code

Algorithme

Codification

Langage de
programmation
(code)

Programme

Interprétation

Exécution par l'ordinateur

Langage machine

La mise en forme d'un algorithme

Algorithme: **nom de l'algorithme**

Le nom doit être significatif, il indique de quoi il s'agit

Variable: **Déclaration des variables et constante**

Constantes:

Début

commentaires

Instruction 1

commentaires

Instruction 2

Fin

Déclaration des Variables, se sont les données d'entrée d'algorithme.

Le corps de l'algorithme

Déclaration des variables et constante

- **Les variables**

Une variable est un objet contenant une valeur et elle est caractérisée par un type. La valeur de cet objet peut à être modifiée au cours de l'algorithme.

- **Comment déclaré une variable en algorithmme ?**

Nom_Variable: Type

- **Qu'ils sont les types possible pour une variable?**

1. **Entier**: les valeurs relatives $n \in \mathbb{Z}$ (exemple: -1, 0, 100).
2. **Réel**: les valeurs réels $x \in \mathbb{IR}$ (0.12, 10.46, -13.5).
3. **Chaine de caractères**: "AC", "1B", "D", "A_#&1".
4. **Caractère** : 'A', '1', '_'.
5. **Booléen**: support deux valeurs vrai (ou bien 1) faux (ou bien 0).

Déclaration des variables et constante

- **Les constantes:**

Des variable dont la valeur est connu et fixe.

- **Comment déclaré une constante en algorithmme ?**

Nom_Variable= Valeur

- **Exemple:**

1. **Pi= 3.14;**

2. **Vitesse_voiture= 60;**

3. **.....**

Exemple

Algorithme: différent type de déclaration variables

Variables:

nom : chaîne de caractères;

nb : Entier;

typePermis : caractère;

prx, mtt, rem : Réel;

verite : Booléen;

Constantes:

Txrem=0.1, pi=3.14;

Début

Fin

Le corps d'algorithme Début.....Fin

Instructions élémentaires

- **Affectation**

L'opération consiste à affecter une valeur à une variable. Elle est représentée par une flèche orientée à gauche.

- $A \longleftarrow 1; \implies A \text{ vaut } 1$
- $B \longleftarrow 3; \implies B \text{ vaut } 3$
- $C \longleftarrow A+B; \implies C \text{ vaut } 4$
- $A \longleftarrow A+3; \implies A \text{ vaut } 4$

Le corps d'algorithme Début.....Fin

Instructions élémentaires

- **Les Opérations Mathématiques:**

Des opérations sur les variables, constantes peuvent être réalisées à l'aide d'opérateurs arithmétiques ou logiques pour former des expressions.

Opération	Operateurs	Exemple
Addition	+	A+1
Soustraction	-	B-A
Multiplication	*	A*C
Division	/	A/C
Le reste de la division	%	A%2
Fonction mathématique	sin(), cos(), exp(), ln(), tan()	sin(A), cos(3.14), exp(A/B),...

Exercices: Affectation

Algorithme: Exercice 1

Variable A, B: Entier

Début:

A ← 1;

B ← A+3;

A ← 3;

Fin

Algorithme: Exercice 2

Variable A, B, C: Entier

Début:

A ← 3;

B ← 10;

C ← A + B;

B ← B + A;

A ← C;

Fin

Exercices: Affectation

Algorithme: Exercice 3

Variable

A, B, C: Entier

D: réel

Constater: $\pi=3.14$;

Début:

A \leftarrow $\cos(2*\pi)$;

B \leftarrow A/2;

B \leftarrow A + B;

C \leftarrow $\sin(A*\pi)$;

C \leftarrow 0;

Fin

Algorithme: Exercice 4

Variable

A, B, C: Entier

D: caractère

Début:

D \leftarrow 'B';

C \leftarrow 5;

B \leftarrow C*10;

C \leftarrow B*C;

Fin

Le corps d'algorithme Début.....Fin

Instructions élémentaires

- **Instruction d'entrée:**

Une instruction d'entrée permet de récupérer une valeur sur un périphérique d'entrée (principalement le clavier).

Notation : **Saisir** (variable); ou bien **Lire**(variable);

- **Instruction de sortie :**

Permet d'afficher à l'écran du "texte", le contenu d'un objet (variable ou constante) voir le résultat d'une expression.

Notation : **Afficher**(variable); ou bien **Ecrire**(variable);

Le corps d'algorithme Début.....Fin

Instructions élémentaires

- **Instruction d'entrée/sortie:**

Algorithme: exemple 1

Variable A, B, C: Entier

Début:

Lire(A);

Lire(B, C);

A <- B;

B <- A;

C <- A + B;

Ecrire("A=",A, "B=",B,"C=",C);

Fin

Algorithme: exemple 2

Variable A, B, C: Entier

Début:

Ecrire("Entrez la valeur de A, B et C");

Lire(A, B, C);

A <- B*C;

B <- C;

C <- A + B;

B <- B + A;

A <- C;

Ecrire("A=",A, "B=",B,"C=",C);

Fin

Exercices: Affectation, Lire, Ecrire..

Exercice 1:

Ecrire un algorithme qui permet de lire votre nom, prénom et âge. Afficher les données entrées

Exercice 2:

Ecrire un algorithme qui demande un nombre, puis qui calcule et affiche le carré de ce nombre.

Exercice 3:

Ecrire un algorithme qui demande le rayon d'un cercle, puis qui calcule et affiche la surface et le périmètre de ce cercle.

Le corps d'algorithme Début.....Fin

Instructions élémentaires

- **Les structures alternatives et conditionnelles**

Permet d'exécuter une instruction ou bien plusieurs instructions suivant une condition ou bien une série de conditions.

Notation 1:

Si (Condition) **alors**

Instruction 1;

Instruction 2;

.....

FinSi

Notation 2:

Si (Condition) **alors**

Instruction 1;

Instruction 2;

.....

Sinon

Instruction 3;

.....

FinSi

Le corps d'algorithme Début.....Fin

Instructions élémentaires

Notation 3:

Si (Condition_1) **alors**

Instruction 1;

.....

Sinon Si (Condition_2) **alors**

Instruction 2;

.....

Sinon Si(Condition_3) **alors**

Instruction 3;

.....

Sinon Instruction 4;

.....

FinSi

FinSi

FinSi

Le corps d'algorithme Début.....Fin

Instructions élémentaires

- Les Operateurs Logiques:**

Comparaison	Operateurs	Exemple
=	Egale à	$A = B$
>	Supérieur à	$A > B$
<	Inferieur à	$A < B$
>=	Supérieur ou égale à	$A >= B$
<=	Inferieur ou égale à	$A <= B$
<>	Diffèrent de	$A <> B$
ET	L'un et l'autre	$A=B$ ET $A<>0$
OU	L'un ou l'autre	$A=B$ OU $A<>0$
NON	Négation de	NON ($A=B$)

Exercices:

Algorithme: Exercice 1

Variable A, B, C: Entier

Début:

A ← 3;

B ← 7;

B ← A + 5;

C ← A + B;

C ← B - A;

Si (C > 10) alors

 A ← B;

 Sinon A ← 5;

FinSi

Ecrire("A=",A,"B=",B,"C=",C);

Fin

Exercices:

Algorithme: Exercice 2

Variable A, B, C: Entier

Début:

A ← 3;

B ← 7;

C ← A + 3;

Si (C > 10 ET B < 5) alors

 A ← B;

 Sinon Si (A <> 0 OU B = A) alors

 C ← A;

 FinSi

FinSi

Ecrire("A=",A,"B=",B,"C=",C);

Fin

Exercices: Affectation, Lire/Ecrire, Si..Sinon

Exercice 1:

Ecrire un algorithme qui permet tester si un entier saisi au clavier est pair ou impair.

Exercice 2:

Ecrire un algorithme qui demande deux nombres, puis affiche le maximum.

Exercice 3:

Ecrire un algorithme qui demande l'âge d'une personne et l'informe ensuite si elle est majeure ou bien mineure.

Le corps d'algorithme Début.....Fin
La structure répétitive

*Comment faire pour Afficher
"Bonjours" 10000 fois ??*

Le corps d'algorithme Début.....Fin

La structure répétitive « **POUR...FinPour** »

La structure **Pour** Permet de répéter un nombre déterminé de fois une (ou plusieurs) action(s).

Notation:

```
Pour (i <- 0; i < N; i <- i+1)
```

```
    inst_1;
```

```
    inst_2;
```

```
    .....
```

```
FinPour
```

```
    inst_3;
```

```
    inst_4;
```

```
    .....
```

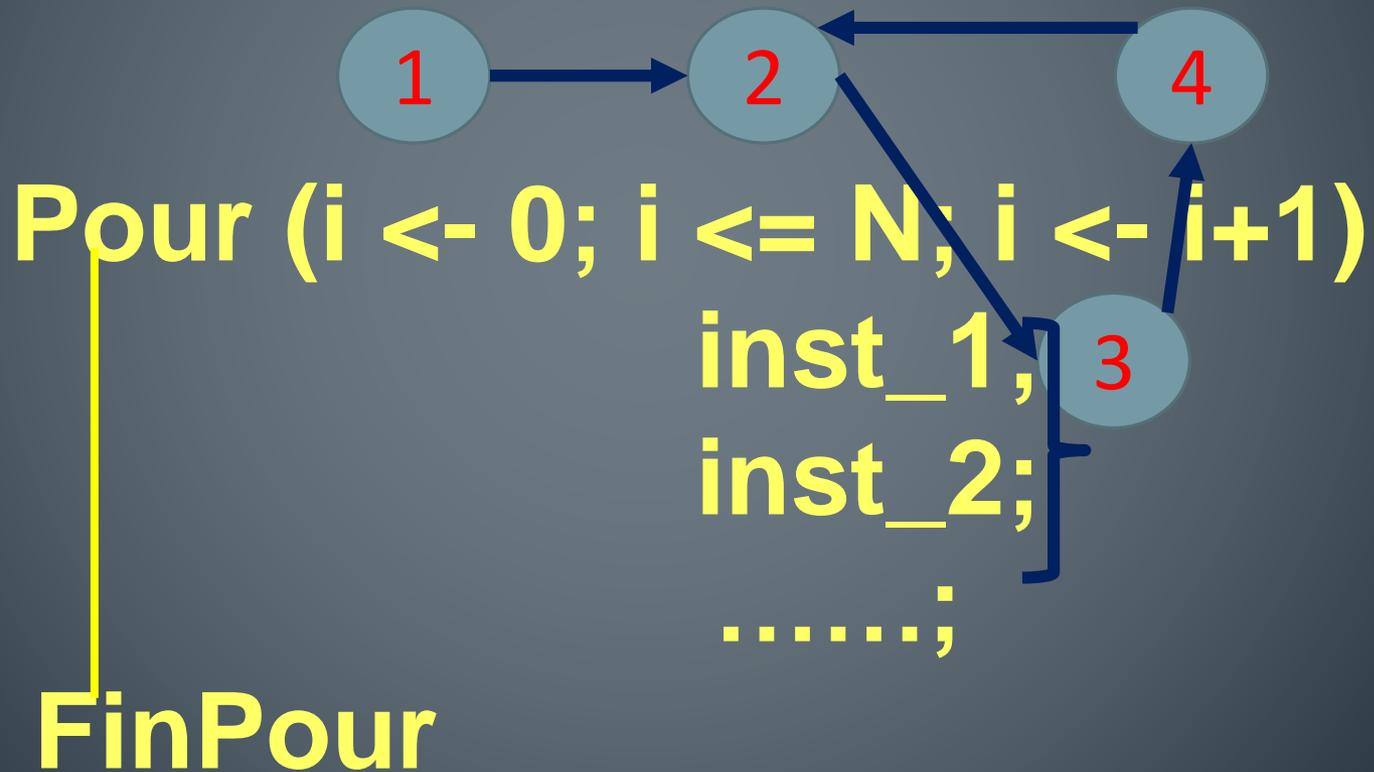
Permet de compter le nombre d'itération des instructions

Lorsque le nombre voulu de répétition est atteint, l'algorithme se poursuit après la boucle

Le corps d'algorithme Début.....Fin

La structure répétitive « **POUR...FinPour** »

En Général: Pour répété une instruction N fois



Exercices:

Exercice 1:

Ecrire un algorithme qui permet d'afficher "Bonjour" 100 fois.

Exercice 2:

Ecrire un algorithme qui permet d'afficher 1 jusqu'à 100.

Exercice 3:

Ecrire un algorithme qui permet d'afficher 100 jusqu'à 1.

Le corps d'algorithme Début.....Fin

La structure répétitive

« TantQue...FinTantQue »

La structure **TantQue... FinTantQue** Permet la répétition d'une (ou plusieurs) action(s) tant qu'une condition est satisfaite.

Notation:

TantQue (condition) **Faire**

instruction_1;

instruction_2;

.....

FinTantQue

*Un nombre déterminé des itérations (**la boucle POUR**).

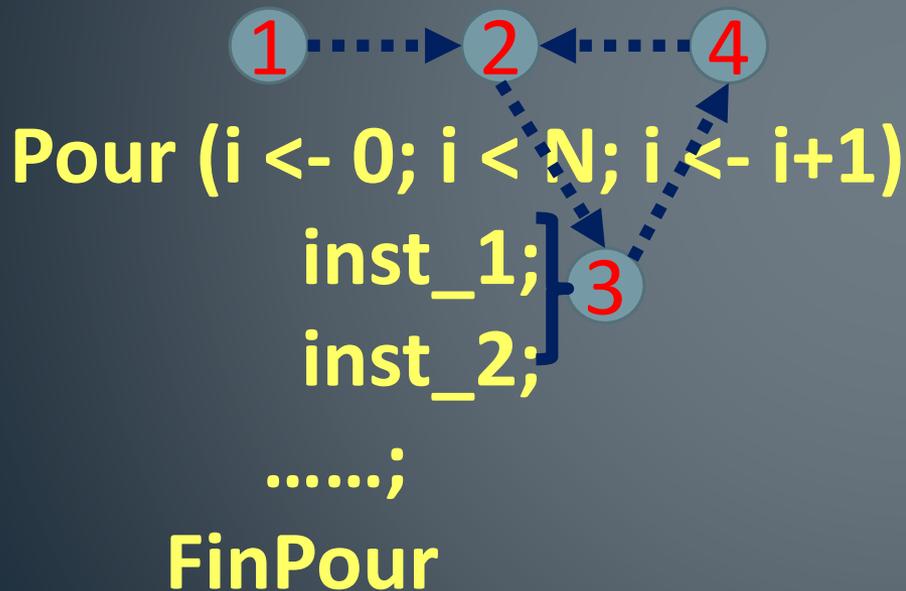
*Un nombre indéterminé des itérations (suivant le condition).

Le corps d'algorithme Début.....Fin

La structure répétitive

« TantQue...FinTantQue »

De la boucle POUR vers la boucle TantQue:



Exercices: a refaire avec la boucle TantQue...FinTantQue

Exercice 1:

Ecrire un algorithme qui permet d'afficher "Bonjour" 100 fois.

Exercice 2:

Ecrire un algorithme qui permet d'afficher 1 jusqu'à 100.

Exercice 3:

Ecrire un algorithme qui permet d'afficher 100 jusqu'à 1.

Le corps d'algorithme Début.....Fin

La structure répétitive « Faire...TantQue »

La structure **Faire... TantQue** Permet la répétition d'une (ou plusieurs) action(s) tant qu'une condition est satisfaite.

Notation:

Faire

instruction_1;

instruction_2;

.....

TantQue (condition);

Différemment de la boucle **TantQue...FinTantQue**, la boucle **Faire...TantQue** exécute les instructions au moins un fois. Soit la condition vraie ou fausse

Exercices: Faire...TantQue

Exercice 1:

Ecrire un algorithme qui permet de saisir et afficher une valeur entre 0 et 100. toutefois, si la valeur saisie par utilisateur n'est pas compris entre 0 et 100, l'utilisateur est obligé de refaire la saisie.

Organigramme

Instructions élémentaires

Début & Fin



Lire() & Ecrire()



Instruction

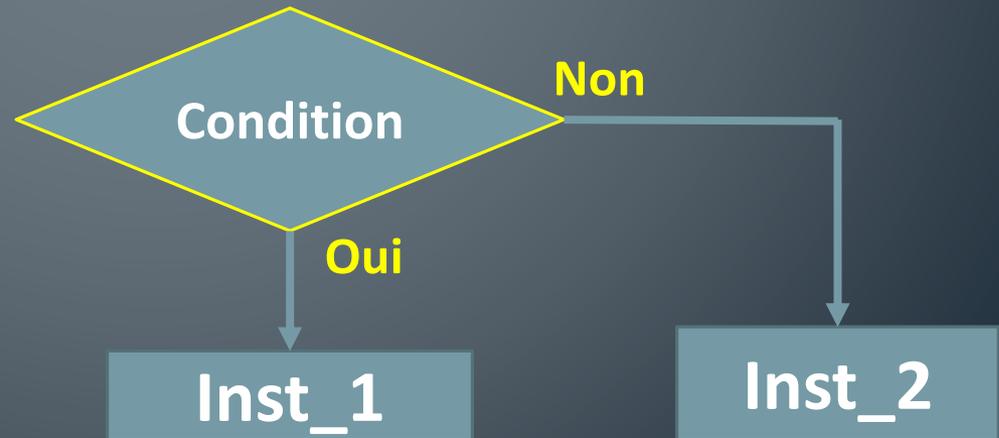


(les opérations en générale)

Si(Condition) alors

inst_1

sinon inst_2;



Organigramme: Exercice

Algorithme: Exercice 2

Variable A, B, C: Entier

Début:

A <- 3;

B <- 7;

C <- A + 3;

Si (C > 10 ET B < 5) alors

A <- B;

Sinon Si (A <> 0 OU B = A) alors

C <- A;

FinSi

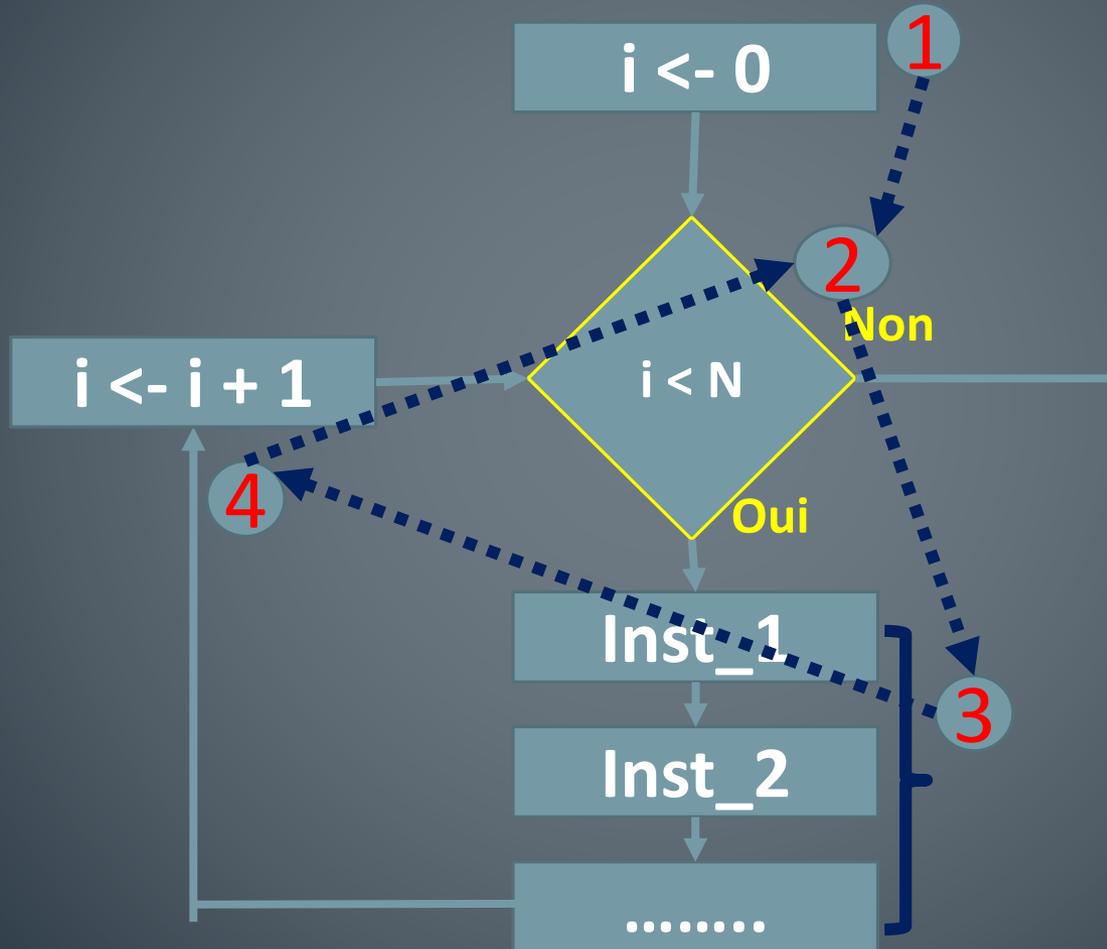
FinSi

Ecrire("A=",A,"B=",B,"C=",C);

Fin

Organigramme

La boucle Pour/TantQue:



Organigramme: Exercice

Algorithme: Exercice 2

Variable i, x: Entier

Début:

x <- 1;

Pour(i <- 1; i < 10; i <- i+1) faire

Si (i % 3 = 1) alors

x <- x + i;

i <- i + i;

Sinon Si (i % 2 = 0) alors

x <- x*2;

FinSi

FinSi

FinPour

Ecrire(x);

Fin

BON COURAGE