

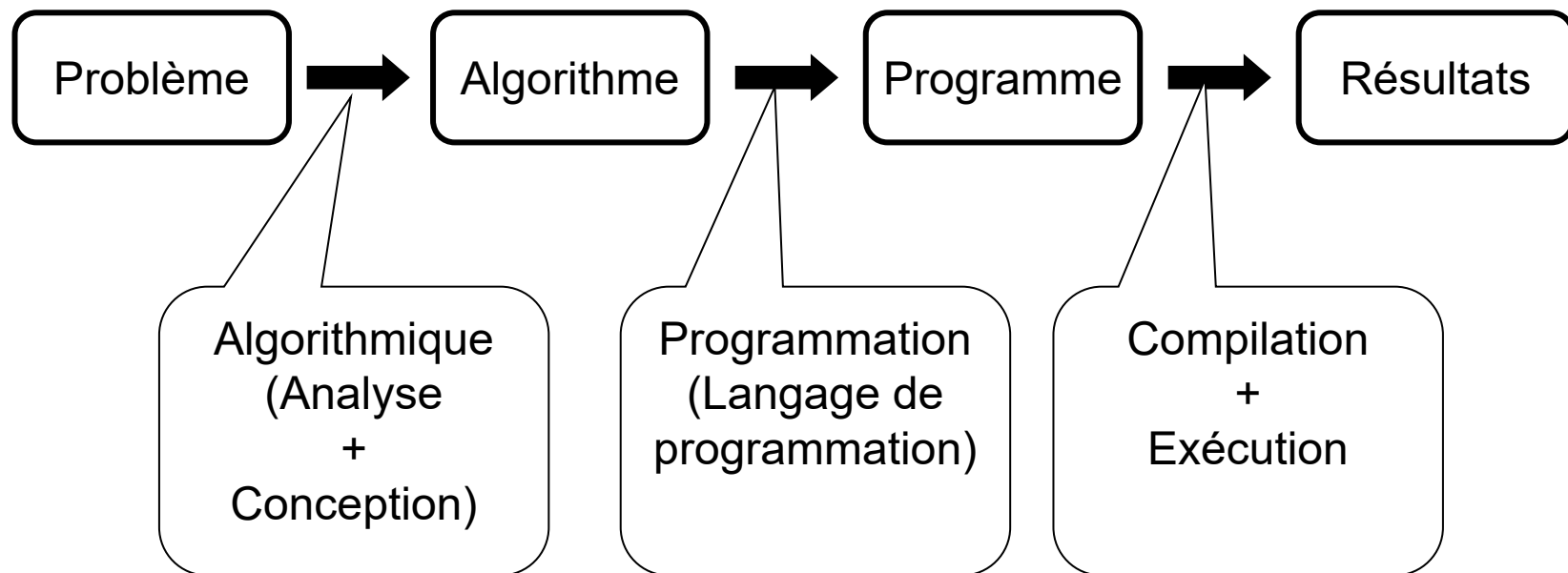


Chapitre 1

Module: Algorithmique & Programmation en C

- 1. Introduction**
- 2. Structure d'un algorithme**
- 3. Instructions de base**
- 4. Organigramme**

Introduction





Introduction (suite)

- **Algorithmique** : consiste à concevoir et mettre au point des algorithmes décrivant les solutions d'un certain type de problèmes.
- L'algorithmique comporte deux grandes phases: (**Analyse et Conception**).
- **Algorithme**: est une suite finie d'instructions à appliquer dans un ordre déterminé sur des données afin d'aboutir à un certain résultat en un temps fini.



Introduction (suite)

- **Analyse:** le dégagement des entrées, des sorties et du traitement.
- **Conception:** l'organisation dans un ordre convenable afin d'aboutir aux résultats attendus.
- **Compilation:** transformation en langage machine d'un programme écrit en langage évolué
- **Exécution:** l'ordinateur exécute les instructions d'un programme en langage « binaire », pour fournir le résultat.



Structure d'un algorithme

Un algorithme peut être structuré en trois parties principales:

- L'en-tête
- La partie Déclarations
- Le corps de l'algorithme



Structure d'un algorithme (suite)

Exemple:

Algorithme Cercle;

Variable Rayon, Surface: **Réel**;

Constante $\text{Pi} \leftarrow 3.14$;

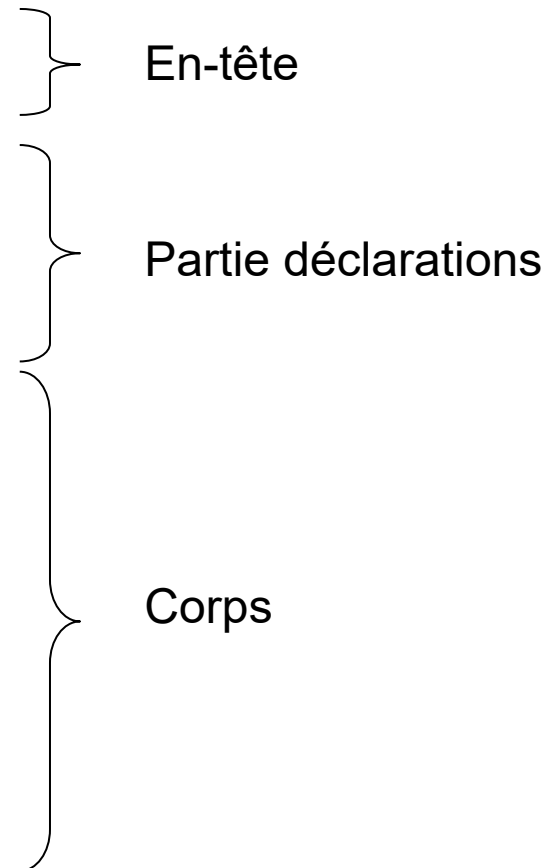
Début

Lire (Rayon);

Surface \leftarrow Rayon*Rayon*Pi;

Ecrire (Surface);

Fin





Instructions de base

- Donnée
- Affectation
- Entrées
- Sorties



Donnée

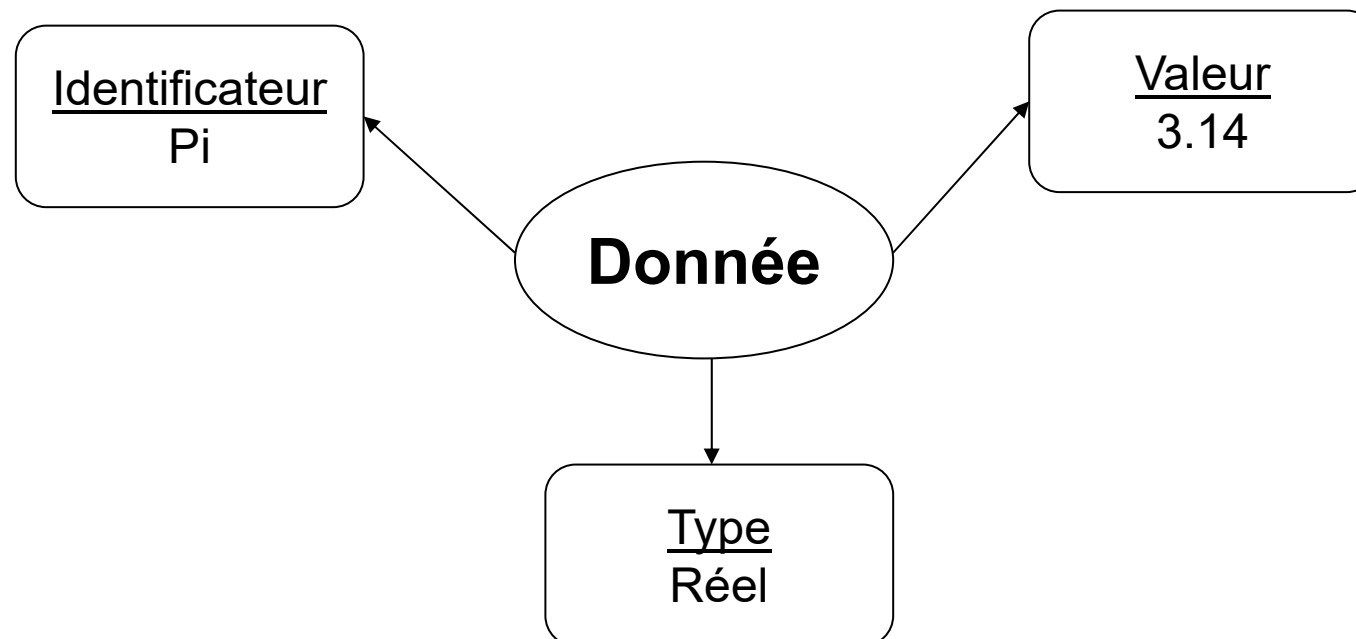
Une donnée est caractérisée par trois attributs:

- **Identificateur**
- **Valeur**
- **Type**



Donnée (suite)

Exemple:





Donnée (suite)

Type (en algorithmique):

<u>Type</u>	<u>Description</u>
Entier	Une donnée de ce type prend ses valeurs dans [-32768,32767]
Réel	Prend ses valeurs dans l'échelle 2.9×10^{-39} à $1.7 \times 10^{+38}$ avec 11 chiffres significatifs après la virgule
Caractère	Prend ses valeurs dans la table des codes ASCII exp 'A' , 'S'
logique	Peut prendre la valeur Vrai ou Faux et on a Faux < Vrai



Donnée (suite)

Une donnée est divisée en deux catégories:

- **Variable:** donnée dont la valeur est susceptible de varier.
- **Constante:** donnée dont la valeur est fixe.



Donnée (algorithmique)

Syntaxe (variable):

variable identificateur_var : **type** ;

Exemple: variable surface : réel;

Syntaxe (constante):

constante identificateur_const ← **valeur** ;

Exemple: constante Pi ← 3.14;



Affectation

- Une affectation est l'opération qui permet de ranger la valeur d'une expression dans une variable.
- Une expression est une combinaison d'opérandes et d'opérateurs.



Affectation (suite)

Quelques opérateurs arithmétiques:

<u>Opérateur</u>	<u>Signification</u>
+	Addition
-	Soustraction
*	Multiplication
/	Division
mod	Reste de la division entière



Affectation (suite)

Quelques opérateurs relationnels:

<u>Opérateur</u>	<u>Signification</u>
=	Egal
<>	Différent
<	Inferieur
>	Supérieur
<=	Inferieur ou égal
>=	Supérieur ou égal



Affectation (suite)

Quelques opérateurs logiques:

<u>Opérateur</u>	<u>Signification</u>
Non OU ET	



Affectation (suite)

Syntaxe (algorithmique):

Identificateur_var ← expression ;

Exemple: Pi ← 3.14 ;

Surface ← Rayon*Rayon*Pi;



Entrées / Sorties

- Instruction d'entrée: acquérir une valeur.
- Instruction de sortie: afficher un message sur l'écran et restituer la valeur d'une variable.



Entrées

Syntaxe (algorithmique):

Lire (identificateur_var);

Exemple: Lire (Rayon) ;



Sorties

Syntaxe (algorithmique):

Ecrire ('message', identificateur_var);

Exemple: Ecrire (Rayon) ;

Ecrire ('Bonjour');

Ecrire ('La somme est:',S);



Exercice 1

Etant donnés le prix hors taxe (PHT) d'un produit et la taxe sur la valeur ajoutée (TVA) 20%.

Ecrire un algorithme qui affiche le prix (TTC).



Exercice 1 (Solution)

Analyse:

- Données d'entrée: PHT
- Données de sortie: TVA, TTC
- Traitement:
 - $TVA = 0.2 * PHT$
 - $TTC = TVA + PHT$



Exercice 1 (Solution)

Conception:

Algorithme Produit;

Variable PHT, TVA , TTC: **Réel**;

Début

Ecrire ('Entrer le prix PHT:');

Lire (PHT);

$TVA \leftarrow 0.2 * PHT;$

$TTC \leftarrow TVA + PHT;$

Ecrire ('Le prix est: ',TTC);

Fin

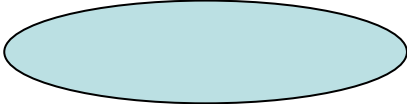


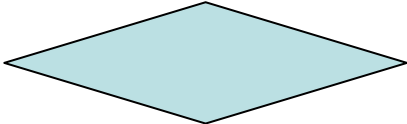



Organigramme

- Un organigramme (ou bien un logigramme) est une représentation graphique de l'algorithme en utilisant un ensemble de figures.



Organigramme (suite)

<u>Figure</u>	<u>Signification</u>
	Début et Fin
	Lire et Ecrire
	Traitement
	Condition
	Sens du traitement



Organigramme (suite)

Exemple:

Algorithme Cercle;

Variable Rayon, Surface: **Réel**;

Constante $\pi \leftarrow 3.14$;

Début

Lire (Rayon);

Surface \leftarrow Rayon*Rayon* π ;

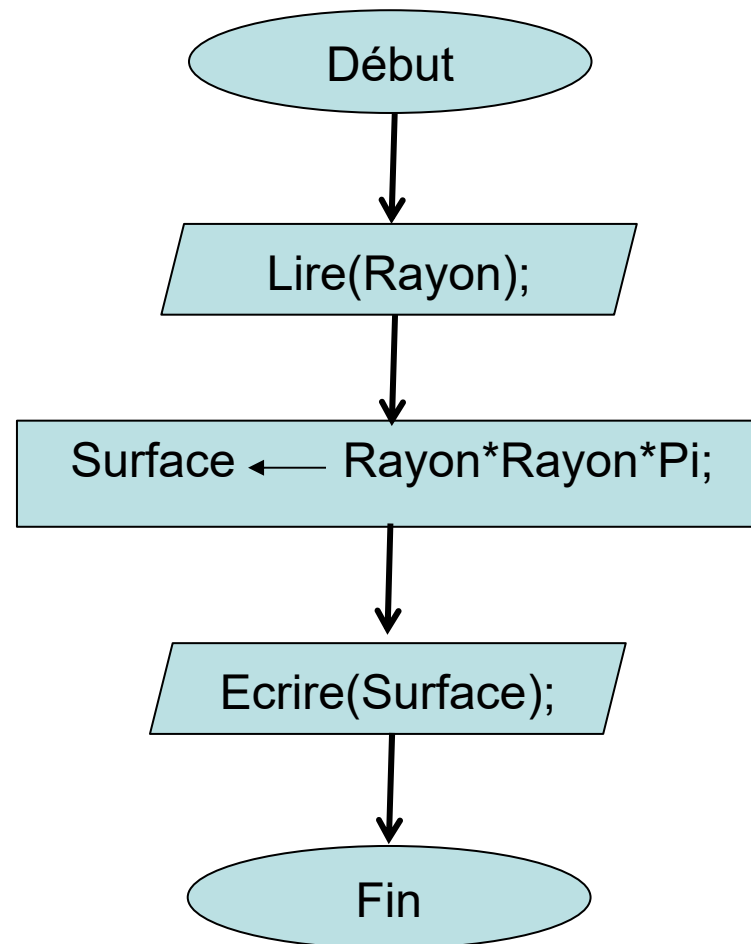
Ecrire (Surface);

Fin

**Donner
l'organigramme
correspondant à
cet algorithme?**



Organigramme (suite)





Exercice 2

Ecrire un algorithme qui calcule et affiche la surface et le périmètre d'un rectangle dont on saisit la longueur et la largeur. On suppose que ces valeurs sont des entiers



Exercice 2 (Solution)

Analyse:

- Données d'entrée: Long, Larg
- Données de sortie: Per, Sur
- Traitement:
 - $Per = 2 * (Long + Larg)$
 - $Sur = Long * Larg$



Exercice 2 (Solution)

Conception:

Algorithme Rectangle;

Variable Long, Larg, Per, Sur: **Entier**;

Début

Ecrire ('Entrer Longueur et Largeur:');

Lire (Long,Larg);

Per $\leftarrow 2 * (Long + Larg)$;

Sur $\leftarrow Long * Larg$;

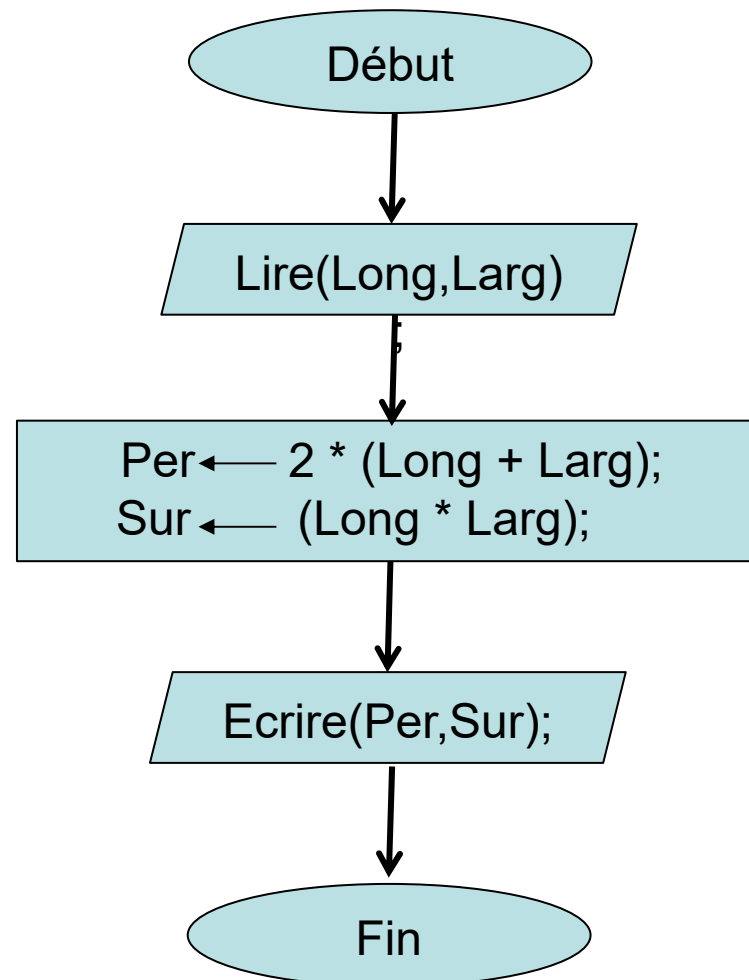
Ecrire ('Périmètre: ',Per);

Ecrire ('Surface: ',Sur);

Fin



Exercice 2 (Solution)





Exercice 3

Ecrire un algorithme qui calcule et affiche la surface (S) et le périmètre (P) d'un cercle dont on saisi le rayon (R).



Exercice 3 (Solution)

Analyse:

- Données d'entrée: R , $P_i=3.14$
- Données de sortie: P , S
- Traitement:
 - $P = 2 * P_i * R$
 - $S = R * R * P_i$



Exercice 3 (Solution)

Conception:

Algorithme Cercle;

Variable R, S, P: **Réel**;

Constante $Pi \leftarrow 3.14$;

Début

Ecrire ('Entrer le rayon:'); **Lire** (R);

$P \leftarrow 2 * Pi * R$;

$S \leftarrow R * R * Pi$;

Ecrire ('Périmètre: ',P);

Ecrire ('Surface: ',S);

Fin



Exercice 3 (Solution)

