Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abderrahmane MIRA de Bejaia

Faculté de Technologie
Département de Technologie



Structure des ordinateurs et applications

À propos du cours

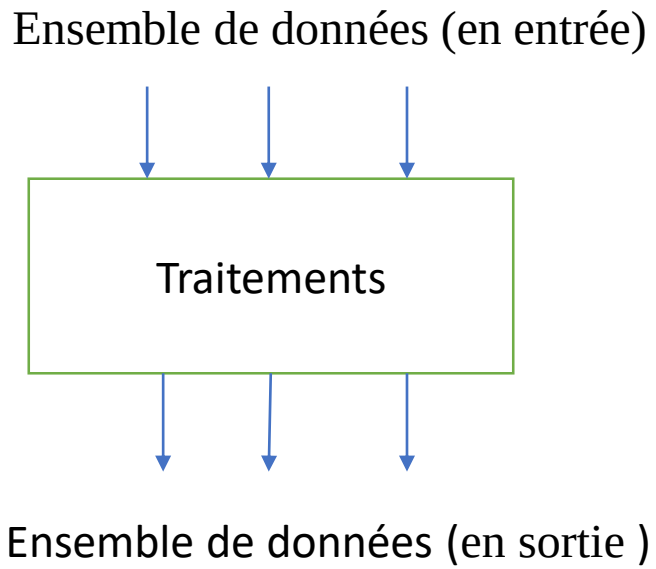
- **Crédits**: 02
- **Coefficients**:02

Notion d'algorithme et de programme

Notion d'algorithme et de programme

- **Concept d'un algorithme**

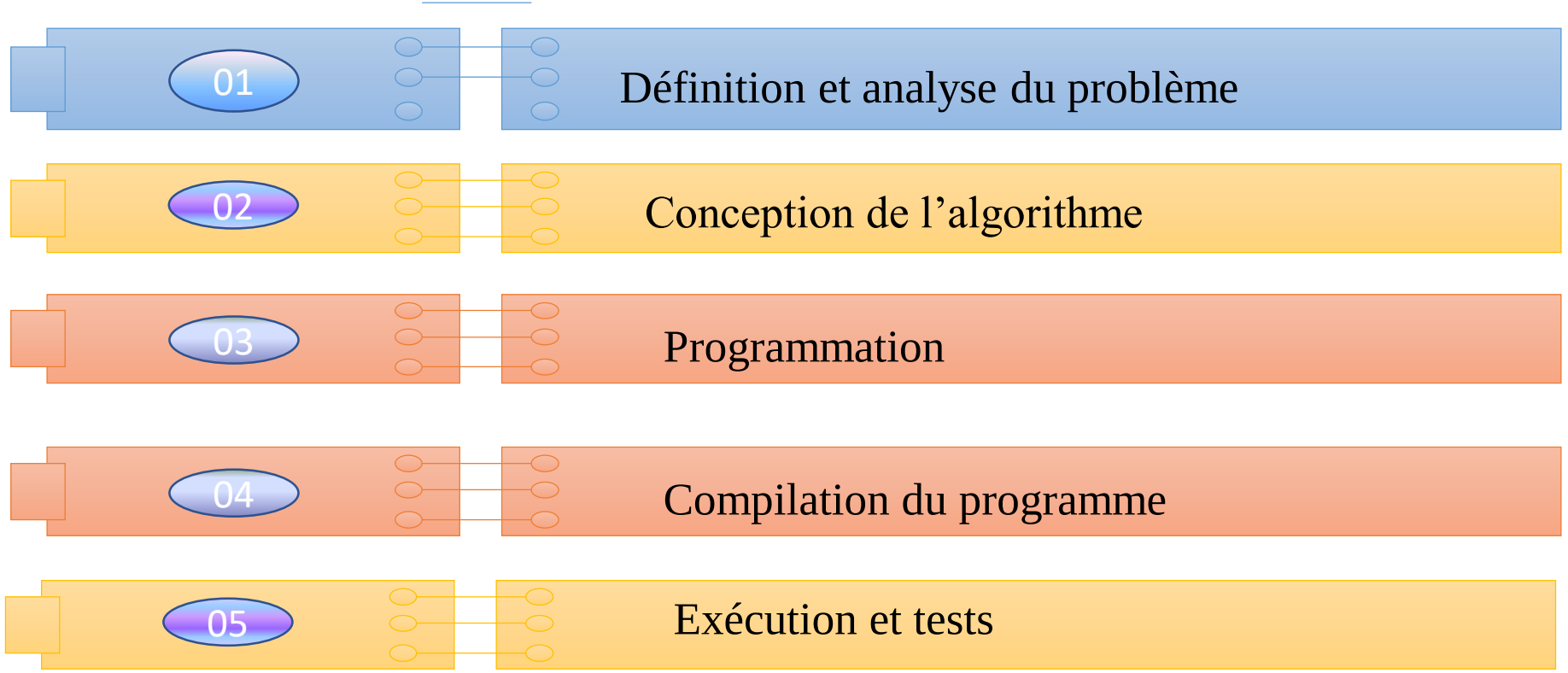
Un Algorithme est une séquence d'instructions ordonnées, qui permet de résoudre un problème. Il reçoit des données en entrée, les traite selon les instructions, et fournit des données en sortie pour obtenir le résultat souhaité.



Notion d'algorithme et de programme

- **Les étapes de résolution d'un problème**

La résolution d'un problème par un algorithme se fait selon plusieurs étapes logiques et ordonnées. Voici les étapes principales à suivre :



Notion d'algorithme et de programme

- **Structure d'un algorithme/programme**

Un algorithme/programme se compose de trois parties principales : L'entête, la partie déclarative et le corps de l'algorithme/programme:

- **L'Entête:** c'est la première ligne de l'algorithme. Elle permet de donner un nom à l'algorithme à travers un identificateur;
- **La partie déclarative :** c'est la section où l'on déclare toutes les données que l'algorithme va utiliser;
- **Le corps de programme/algorithme :** c'est l'ensemble des instructions qui font fonctionner l'algorithme.

Notion d'algorithme et de programme

- **L'Entête**

L'entête sert à donner un nom à l'algorithme en utilisant un identificateur. Ce dernier est précédé par le mot clé "**Algorithme**". Alors qu'est-ce qu'un identificateur ?

Identificateur : Un identificateur est une chaîne de caractère qui permet de donner un nom unique à un programme (algorithme), une constante ou une variable. Cette chaîne doit commencer soit par un caractère alphabétique ou par un tiret du 8 (_) et ne peut contenir que des caractères alphanumériques. Aussi, les mots réservés (mots-clés) d'un langage de programmation ne peuvent être utilisés comme identificateurs.

Identificateur valide	Identificateur non valide
Commencer soit par un caractère alphabétique ou par un tiret du 8 (_)	Commencer soit par un caractère spécial ou par un caractère numérique
	Contenir un caractère special &,%,$,#, @,?,-:,.+
Contenir que des caractères alphanumériques	(mots-clés) réservés au langage C: *auto, break, case, char, continue, do, double, else, extern, float, for, goto, if, int, long, return short, sizeof, static, struct, switch, typedef, union, unsigned, while, …*

Tableau II.3 : Exemples d'identificateurs valides et non valides

Identificateur valide	Identificateur non valide
a1	x1 y (à cause du blanc ou l'espace)
a_1	x1-y (à cause du signe –(tiret de 6))
A_1	1xy (commence par un caractère numérique)
x12y	R? (à cause de caractère spécial (?))
x1_y	Prix-HT (à cause du signe – (tiret de 6))
Hauteur	Exo 04 (à cause de l'espace)
Prix_HT	Program (mot clé)
Exo_04	Read (mot clé)

Notion d'algorithme et de programme

Structure d'un algorithme/program

Algorithme	Langage C
Algorithme <identificateur_algo> ;	#include <stdio.h>
	int main (){
Constantes <identificateur_constante> = <valeur> ;	// Section de déclaration
Variables <identificateur_variable> : <Type> ;	<TypeDonnée> <identifiant> ;
	Const <Type> <identifiant> = <Valeur> ;
Début <Instruction 1> ; <Instruction 2> ; ... <Instruction n> ;	<Instruction 1> ; <Instruction 2> ; ……
Fin.	return 0 ;

L'Entête

La partie déclarative

Le corps de programme



C'est quoi une bibliothèque en langage C ?

Une bibliothèque est un ensemble de fonctions déjà prêtes qu'on peut utiliser dans notre programme au lieu de les écrire nous-mêmes

Bibliothèque	Utilité	Exemple
stdio.h	Entrées et sorties	printf(), scanf()
stdlib.h	Mémoire, conversions, fonctions diverses	malloc(), free(), rand()
string.h	Manipulation des chaînes de caractères	strlen(), strcpy(), strcmp()
math.h	Calculs mathématiques	sqrt(), pow(), sin()
stdbool.h	Utilisation du type booléen	bool, true, false

Important

int : indique que la fonction retourne un nombre entier.

main() : c'est la fonction principale, le point de départ du programme.

{ Indique le début du bloc d'instructions de la fonction main().

return 0; Indique que le programme s'est terminé normalement.

}Indique la fin du bloc d'instructions de main().

scanf() : lire une valeur saisie par l'utilisateur.

printf(): permet d'afficher un message ou une valeur à l'écran.

une variable intermédiaire est généralement une variable qui stocke un résultat temporaire nécessaire à un calcul ultérieur.

scanf() sert à lire une valeur et à la placer dans une variable. Il a donc besoin de son adresse mémoire : &N.

printf() sert à afficher une valeur. Il a seulement besoin de la valeur elle-même : N

Notion d'algorithme et de programme

- **La partie déclarative**

La partie déclarative sert à déclarer les différentes données que l'algorithme utilise (Constantes, variables,.. etc.). Une donnée non déclarée et utilisée par le programme engendre une erreur lors de la compilation. Alors qu'est-ce qu'une variable et qu'est-ce qu'une constante ?

- **Constantes** c'est une donnée qui ne change jamais pendant l'exécution du programme;

En algorithme:

Constantes <identifiant> = <valeur> ;

En langage C:

const <type> <identifiant> = <valeur> ;

Exemple :

Tableau II.4 : Exemples de constantes

Algorithme	Langage C
const PI = 3.14; Constante réelle. **const** MAX = 10; Constante entière. **const** cc = 'a'; Constante caractère. **const** ss = 'algo'; Constante chaîne de caractère. **const** b1 = true; Constante booléenne. **const** b2 = false; Constante booléenne.	**const** float PI = 3.14; **const** int MAX = 10; **const** char cc = "a"; **const** char ss[10] = "algo"; **const** int b1 = 1; . **const** int b2 = 0;

Variables :est un espace mémoire qui contient une valeur pouvant être modifiée. Les variables sont déclarées comme suit :

En algorithme

<identifiant> : <TypeDonnée>;

En langage C:

<TypeDonnée> <identifiant>;

Une variable appartient à exactement un type de données. Il existe cinq types de données fondamentaux :

Tableau II.5 : Types de Données Fondamentaux et leur Déclaration en Algorithme et en C

Type de Données	Algorithme	Langage C	Exemples
Entier	x : Entier;	int x;	…, -2, -1, 0, 1, 2, …
Réel	x : Réel;	float x; **ou** double x;	3.14, -0.5, 2.718
Caractère	c : Caractère;	char c;	'A', '7', '@'
Chaîne de caractères	s : Chaîne;	char s[50];	"Bonjour", "CS101"
Booléen	b : Booléen;	int b; **ou** bool b;	VRAI / FAUX → 1 / 0

Exemple :

Tableau II.6 : Exemples de variables

Algorithme	Langage C	Signification
x : réel ; n, m : entier ; s : chaîne de caractères ; b1, b2, b3 : booleen ; c1 : caractère ;	float x ; int n, m ; char s[100]; bool b1, b2, b3; char c1;	variable réelle deux variables entières variables chaîne de caractères 3 variables booléennes variable caractère

Remarque : En plus des constantes et des variables, il est également possible de déclarer de nouveaux types, fonctions et procédures

Chaque instruction utilisée en algorithme possède une écriture équivalente en langage C.

Tableau II.7: Correspondance Algorithme/Langage C

Algorithme	Langage C	Spécificateur
Entier	int	%d
Réel	float	%f
Booléen	bool	%d
Caractère	char	%c
Chaîne	char[ ]	%s

Un spécificateur de format en langage C est un symbole qui indique le type de donnée à lire ou à afficher avec scanf ou printf.

Remarque : Le type bool nécessite **#include <stdbool.h>**. 1 représente VRAI (TRUE) et 0 représente FAUX (FALSE).

II.5.3. Le corps de programme

Le corps d'un algorithme est constitué d'un ensemble d'actions / instructions ordonnées de manière séquentielle et logique. Ces instructions se divisant en cinq types distincts:

- ***Lecture*** **:** Cette opération consiste à introduire des données dans l'algorithme. Une lecture consiste à donner une valeur arbitraire à une variable.
- ***Écriture*** **:** Cette opération implique l'affichage de données. Elle permet d'afficher des résultats ou des messages.
- ***Affectation*** **:** Elle permet de modifier les valeurs des variables en leur assignant de nouvelles valeurs.
- ***Structures de contrôle*** **:** Ces structures permettent de modifier la séquentialité de l'algorithme. Elles sont utilisées pour sélectionner différents chemins d'exécution ou pour répéter un traitement.

Dans le langage C, chaque instruction se termine par un **point-virgule**.

Notion d'algorithme et de programme

- **Le corps de programme(Types d'instructions)**

1. **Instructions d'Entrées/Sorties (Lecture / Écriture)**

Entrées (Lecture)

Une instruction d'entrée est une instruction qui permet d'introduire (saisir) une valeur à l'Algorithme à travers le clavier.

La syntaxe d'une lecture

Algorithme	Langage C	Signification
Lire(<id_var>);	scanf ("%c", &<id_var>);	Donner une valeur quelconque à la variable dont l'identifiant *<id_var>*.
Lire(<iv1>, <iv2>, …);	scanf (" %c %c ….", &<iv1> , &<iv2>,…);	Donner des valeurs aux variables <iv1>, <iv2>, *etc.*

Notion d'algorithme et de programme

- **Le corps de programme(Types d'instructions)**

1. **Instructions d'Entrées/Sorties (Lecture / Écriture)**

Sorties (Écriture))

L'écriture est une instruction qui permet d'afficher, à l'écran, des données. Ces dernières peuvent être un message, une valeur(nombre), la valeur d'une variable, une constante et même le résultat d'un calcul.

La syntaxe d'une écriture

Algorithme	Langage C	Signification
Ecrire(<id_var> | <id_const> | <valeur>, | <expression>)	printf("%f", <id_var> |<expression>);	Afficher une valeur d'une variable, d'une constante, valeur immédiate ou calculée à travers une expression.

Instruction d'affectation

Une affectation consiste à donner une valeur (immédiate, constante, variable ou calculée à travers une expression) à une variable.

Algorithme	Langage C
<id_varialbe> ← <valeur> | <id_variable> | <expression> ;	<id_varialbe> = <valeur> | <id_variable> | <expression> ;

Une affectation possède deux parties : la partie gauche qui représente toujours une variable, et la partie droite qui peut être : une valeur, variable ou une expression. La condition qu'une affectation soit correcte est que : la partie droite doit être du même type (ou de type compatible) avec la partie gauche.

Notion d'algorithme et de programme

- **Le corps de programme(Types d'instructions)**

Remarque :

Pour commenter un programme en langage C, on écrit les commentaires entre /*…. */. Par exemple : ***/*Ceci est un commentaire*/***

- On peut aussi utiliser les commentaire ligne :

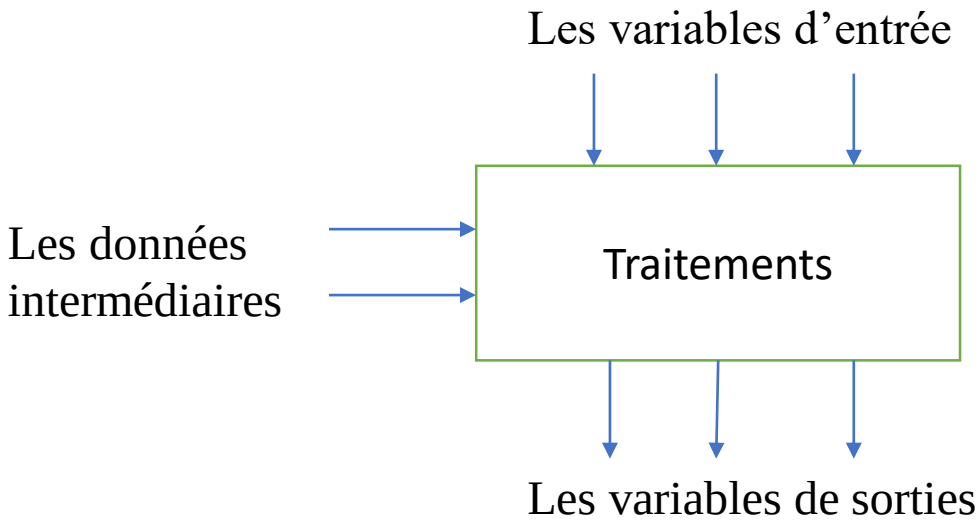
// Ceci est un commentaire ligne

- En langage C, la partie déclarative peut être avant la fonction main, comme peut être dans la fonction main.
- En langage C, la fonction **main** est le point d'entrée du programme : la première qui sera exécutée.

Le **commentaire** n'est pas pris en compte à la compilation. Il sert à rendre le programme plus clair à la lecture, à noter des remarques, etc.

Notion d'algorithme et de programme

- **Schéma générale d'un algorithme**



- Entrées (Inputs) : les données initiales nécessaires pour démarrer le traitement. Ce sont généralement les **variables lues** avec l'instruction Lire;
- **Traitement (Process) :** ce sont les opérations ou calculs effectués sur les données d'entrée
- **Les données intermédiaires** ce sont des valeurs utilisées pendant le traitement, comme des variables ou des constantes, qui ne sont ni lues ni affichées directement ;
- **Sorties (Outputs) :** ce sont les résultats produits par l'algorithme, généralement affichés à l'écran avec l'instruction Écrire.

Exemple 1: (Conversion d'Euros en Dinars)

Écrire un programme en langage Cqui permet de convertir un montant en euros en dinars.

Le programme devra :

1. Demander à l'utilisateur de saisir le montant en euros;
2. Utiliser un taux de conversion fixe (par exemple : 1 Euro = 16 Dinars);
3. Calculer et afficher le montant correspondant en dinars..

Exemple 2: (Calcul de la moyenne de trois nombres)

Écrire un programme langage C qui :

1. Lit trois nombres réels A, B, C;
2. Calcule la moyenne M = (A + B + C) / 3;
3. Affiche la moyenne.