



-Le Langage Algorithmique-

<https://goo.gl/J4R515>

Exemples de Problème :



Solution :

Outils:

Un Cric - Une Clé à croix - Une Roue de Secours

Etapas :

1. Serrer le frein à main de la voiture.
2. Desserrer les boulons d'un seul tour à l'aide de la clé en croix. Dans le sens inverse des aiguilles d'une montre.
3. Placez le cric sous la voiture.
4. Levez la voiture jusqu'à ce que le pneu soit au moins à 5 cm du sol
5. Dévissez les boulons et retirez la roue crevée.
6. Placez la roue de secours et vissez les boulons.
7. Rabaissez la voiture.
8. Enlever le cric.

ALGORITHME

4

INTRODUCTION

2

Exemples de Problème :

Problème 1:

Pneu crevé. Comment changer la roue d'une voiture?



3

Exemples de Problème :

Problème 2:

On désire faire un gâteau à partir des ingrédients suivants.

- Sucre.
- Farine.
- Œufs.
- Chocolat
- Beurre.
- Levure.



5

Exemples de Problème :



Solution :

Ingrédients

200g de chocolat - 150g de sucre - 4 oeufs - 50g de farine - 150g de beurre - 1/2 paquet de levure

Préparation:

1. Faire fondre le **chocolat** avec le **beurre**
2. Séparer les blancs des jaunes d'**œufs**.
3. Battre les blancs.
4. Mélangez la préparation **chocolat** avec le **sucre**
5. Ajoutez les jaunes d'**œufs**.
6. Ajoutez la **farine**.
7. Ajoutez la **levure**.
8. Ajoutez les blancs d'**œufs**.
9. Mettre le gâteau pour 30 min de cuisson.

ALGORITHME

6

Exemples de Problème :

Problème 3:

Trouver la valeur de x tel que : $ax^2 + bx + c = 0$. ($a, b, c \neq 0$)

Solution :

- $\Delta = b^2 - 4ac$.
- Si $\Delta > 0$ Deux Solutions : $x_0 = \frac{-b - \sqrt{\Delta}}{2a}$; $x_1 = \frac{-b + \sqrt{\Delta}}{2a}$
- Si $\Delta = 0$ Une Solution : $x_0 = \frac{-b}{2a}$
- Si $\Delta < 0$ Aucune Solution.

ALGORITHME

7

DEFINITIONS

8

Quelques Définitions :

Un Algorithme :

Un Algorithme est une suite ordonnée d'instructions qui indique la démarche à suivre pour résoudre un problème.

L'Algorithmique :

L'algorithmique est la science qui étudie l'application des algorithmes à l'informatique.

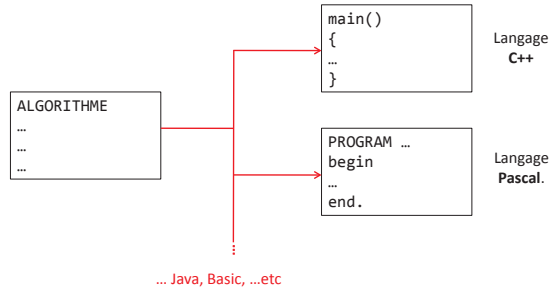
L'informaticien traduit l'algorithme en un langage compréhensible par un ordinateur. (Langage de programmation).

9

Quelques Définitions :

Un Programme :

Un Programme est la traduction d'un algorithme en un langage de programmation.



10

Quelques Définitions :

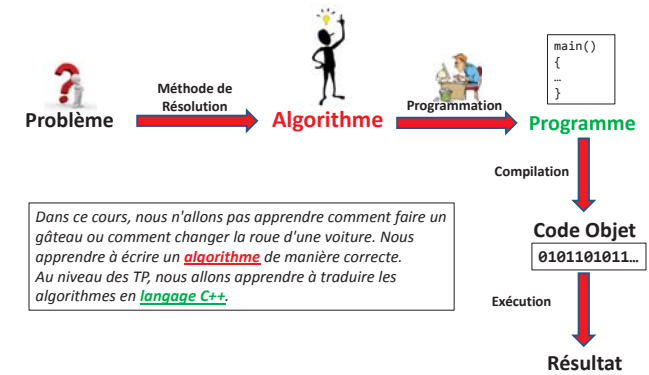
La Compilation :

La Compilation est la traduction d'un programme écrit en langage de programmation (Ex: C, C++) en un langage pouvant être exécuté par un ordinateur (Code Objet).



11

Etapes de Résolution d'un problème :



12

LES VARIABLES

13

Les Variables.

Un ordinateur est capable de Stocker et de traiter les informations.

Un algorithme doit pouvoir accéder aux informations stockées dans la mémoire. Pour cela on utilise des **Variables**.

Une **variable** est une information stockée de manière temporaire dans la mémoire principale. La valeur d'une variable peut évoluer au cours de l'exécution de l'algorithme d'où le terme "variable".

Une Variable est caractérisée par :

- **Un Identifiant** : Le nom donné à la variable.
- **Un type** : C'est l'ensemble des valeurs que peut avoir la variable.
- **Une valeur** : C'est la valeur que stocke la variable.

14

Les Variables.

Exemple 1:

On désire faire un algorithme qui demande à l'utilisateur la longueur et la largeur d'un rectangle et calcul la surface du rectangle.
De quelles variables avons-nous besoin?

Solution :

On a besoin de trois variables :

- Une variable nommée **Long** qui contient la longueur du rectangle.
- Une variable nommée **Larg** qui contient la largeur du rectangle.
- Une variable nommée **Surf** qui prendra la valeur de la surface du rectangle.

15

Les Variables.

Exemple 2:

On désire écrire un algorithme qui résout une équation du second degré.
De quelles variables avons-nous besoin?

$$ax^2 + bx + c = 0$$

Solution :

On a besoin de :

- Trois variables (nommées **a, b, c**) contenant les coefficients de l'équation
- Une Variable (nommée **delta**)
- Deux variables pour les solutions (nommées **x0** et **x1**)

16

Les Variables.

L'identifiant :

L'identifiant (le nom) d'une variable peut être une combinaison de chiffre et de lettres mais doit respecter les conditions suivantes :

- Un Identifiant doit impérativement commencer par une lettre.
- Un Identifiant ne peut pas contenir des caractères accentués (ex : é, ê, ë).
- Un Identifiant ne peut pas contenir des caractères spéciaux (@ #, ...etc). Seul le caractère "_" (underscore) est autorisé,
- Un Identifiant ne peut pas contenir le symbole " " (espace).
- Un Identifiant ne peut pas être un **mot clé**. (nous allons voir ça plus tard...)

17

Les Variables.

Types d'une Variable

Toute variable doit avoir un **type** qui représente l'ensemble des valeurs que peut contenir la variable.

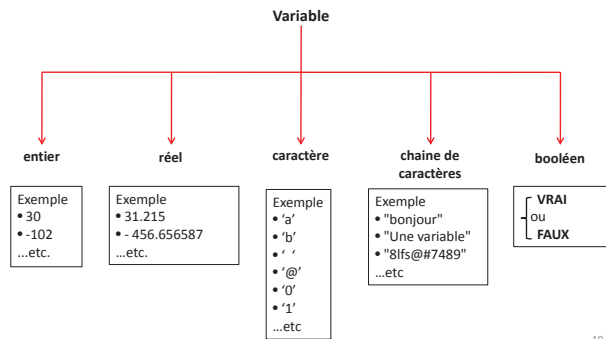
En algorithmique, il existe 5 types élémentaires :

- Le type **entier** : Représente l'ensemble des entiers relatifs (ex : 45 2 -458,...etc).
- Le type **réel** : Représente l'ensemble des valeurs réelles (ex : 12.54 -45.6584 13 -45.32 ... etc.)
- Le type **caractère** : peut être une lettre ('A', 'a', ...etc), une chiffre ('0','1',...,'9') , un symbole ('&', '#', '@', '(', ...etc) ou un espace (' ')
- Le type **chaîne de caractère** : suite de caractères ("Mohamed", "x #@"...etc).
- Le type **booléen** : ne peut avoir que deux valeurs : **VRAI (true)** ou **FAUX (false)**

18

Les Variables.

Types d'une Variable



19

Opérations sur les Variables.

Les Entiers :

Les opérations autorisées sur les entiers sont :

- L'addition (+).
- La Soustraction (-).
- La multiplication (*).
- La division entière (div) :
Exemple :
12 div 5 = 2
19 div 5 = 3
- La division (/) :
Exemple :
12 / 5 = 2
19 / 5 = 3
- Le modulo (mod) : Représente le reste de la division.
Exemple :
15 mod 6 = 3
30 mod 4 = 2

20

Opérations sur les Variables.

Les Réels :

Les opérations autorisées sur les réels sont :

- L'addition (+).
- La Soustraction (-).
- La multiplication (*).
- La division (/) :
Exemple :
12.0 / 5.0 = 2.4
19.0 / 5.0 = 3.8

21

Opérations sur les Variables.

Les Chaînes de Caractères:

- La concaténation :
La concaténation consiste à ajouter une chaîne à la fin d'une autre chaîne.

- Les chaînes de caractères sont mise entre guillemets " " afin de les distinguer des valeurs numériques ou des variables.
- Les caractères sont mise entre apostrophes : ' '.

22

Opérations sur les Variables.

Les Booléens :

Nous avons principalement trois types d'opérations logiques pour les booléens :

- ET En C++ : &&
- OU En C++ : ||
- NON En C++ : !

A	B	A ET B
FAUX	FAUX	FAUX
FAUX	VRAI	FAUX
VRAI	FAUX	FAUX
VRAI	VRAI	VRAI

Opérateur ET

A	B	A OU B
FAUX	FAUX	FAUX
FAUX	VRAI	VRAI
VRAI	FAUX	VRAI
VRAI	VRAI	VRAI

Opérateur OU

A	NON A
FAUX	VRAI
VRAI	FAUX

Opérateur NON

23

Opérations sur les Variables.

Les Booléens :

Une expression booléenne peut être conçu à l'aide d'opérateurs logiques (ET, OU, NON) et d'opérateurs de comparaisons.

Opérateur	Algorithmique	C++
Egal à:	=	==
Est différent de:	≠	!=
Supérieur à:	>	>
Supérieur ou égal à:	≥	>=
Inférieur à:	<	<
Inférieur ou égal à:	≤	<=

Exemple :

Algorithme
((x<5) ET (y<>5)) OU (z=3)

C++
((x<5) && (y!=5)) || (z==3)

24

Déclaration de Variables

La déclaration d'une variable sert à réserver un espace de stockage dans la mémoire principale.

Toutes les variables utilisées dans un algorithme doivent être déclarées

Une variable est déclarée par la syntaxe suivante :

Identifiant_Variable : Type ;

Exemple :

i : Entier ;
J : Entier ;
X : Réel ;

IL EST STRICTEMENT INTERDIT DE DECLARER PLUS D'UNE FOIS LA MÊME VARIABLE

~~i : Entier ;~~
~~J : Entier ;~~
~~i : Entier ;~~

25

Déclaration de Variables

Il est possible de déclarer plusieurs variables de même type en une seule ligne en utilisant la syntaxe suivante :

IdV1 , IdV2 , ... , IdVn : Type ;

Exemple :

i , j , k : Entier ;
C1 , C2 : Caractère ;
Ch1 , Chaîne2 : Chaîne de caractères.

26

Déclaration de Variables

Les Constantes :

Une constante est une variable dont le contenu ne change pas.

Une Constante est déclarée comme suit :

const Identifiant_Constante = Valeur ;

Exemple :

const PI = 3.141592 ;

27

Affectation de Variables

Lorsqu'une variable est déclarée, elle ne contient initialement aucune valeur.

Pour donner une valeur à une variable on utilise l'instruction d'**affectation**.

La syntaxe est la suivante :

```
IdVariable ← Expression ;      En C++:
                                IdVariable = Expression ;
```

Dans cette instruction, le contenu de l'**expression** est mis dans la variable **IdVariable**.

L'expression et la variable doivent avoir le même type. (ou compatibles)

L'expression peut être :

- Une Constante.
- Une Variable.
- Une Combinaison de Variables et d'opérations.

28

Affectation de Variables

Exemples :

1-) Affectation par une constante

```
x ← 5 ;
```

Après exécution de l'instruction la variable **x** aura la valeur 5 (x=5).

2-) Affectation par une variable

```
x ← y ;
```

Après exécution de l'instruction la variable **x** aura la valeur de la variable **y** (x=y)
La variable **y** reste inchangée.

3-) Affectation par une expression

```
x ← (y+z)/2 ;
```

Après exécution de l'instruction la variable **x** aura comme valeur, la moyenne des variables **y** et **z**.

29

STRUCTURE D'UN ALGORITHME

30

Syntaxe d'un Algorithme

Un algorithme doit respecter la syntaxe suivante :

```
Algo Id_Algo
Début
Var /* Déclarations de Variables*/

//Bloc d'instructions

FIN.
```

Ou

```
Algo Id_Algo
Var /* Déclarations de Variables*/
Début

//Bloc d'instructions

FIN.
```

Id_Algo est l'identifiant (le nom) de l'algorithme.

31

Syntaxe d'un Algorithme

Exemple 1 :

```
Algo Somme
Début
Var X,Y,Z : Réel;
X ← 5 ;
Y ← 8 ;
Z ← X + Y;
FIN.
```

Quelles sont les valeurs des variables **X,Y** et **Z** après exécution de l'algorithme **Somme** ?

32

Syntaxe d'un Algorithme

Exemple 2 :

```
Algo Moyenne1
Début
Var X,Y,Z : Réel;
X ← 5 ;
Y ← 8 ;
Z ← (X + Y) / 2;
FIN.
```

Quelles sont les valeurs des variables **X,Y** et **Z** après exécution de l'algorithme **Moyenne1** ?

33

Syntaxe d'un Algorithme

Exemple 3 :

```
Algo Moyenne2
Début
Var X,Y,Z : Entier ;
X ← 5 ;
Y ← 8 ;
Z ← (X + Y) div 2 ;
FIN.
```

Quelles sont les valeurs des variables **X,Y** et **Z** après exécution de l'algorithme **Moyenne2** ?

34

Les Commentaires

Les commentaires sont des textes écrits à l'intérieur de l'algorithme qui ne sont pas pris en compte.

Les commentaires ont pour but d'expliquer une partie du programme afin d'améliorer sa lisibilité et faciliter sa compréhension.

En C++ Pour mettre un commentaire, on utilise les symboles **// ou /* */**

```
Algo Moyenne1
Var
X,Y,Z : Réel ;
Début
X ← 5 ;
Y ← 8 ;
Z ← (X + Y) / 2 ; // Z prend la moyenne de X et Y.
FIN.
```

35

LES ENTREES / SORTIES

36

Les Entrées / Sorties

Exemple :

Soit l'algorithme suivant :

```
Algo Surface
Var Long , Larg , Surf : Réel ;
Début

    Long ← 25;
    Larg ← 10 ;
    Surf ← Long * Larg ;

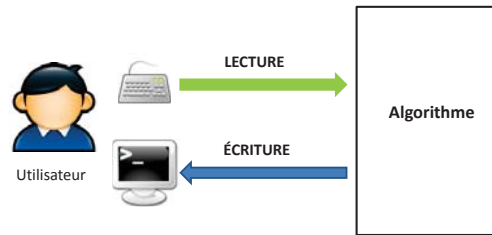
FIN.
```

Problèmes :

1. Cet algorithme calcule uniquement la surface d'un seul rectangle (25X10).
2. Le résultat du calcul n'est pas affiché.

37

Les Entrées / Sorties



On souhaite que :

- L'utilisateur donne lui-même à l'aide du clavier, les valeurs de la longueur et la largeur.
- L'algorithme affiche le résultat du calcul (La valeur de la surface) sur l'écran.

38

Les Entrées / Sorties

Un Algorithme a souvent besoin de communiquer avec l'utilisateur. Pour cela il existe deux types d'instruction

LECTURE : Permet de lire une valeur que fournit l'utilisateur à l'aide du clavier.

ÉCRITURE : Permet d'afficher une valeur à l'écran.

39

La Lecture :

La lecture est réalisée à l'aide de l'instruction **Lire**, sa syntaxe est la suivante :

```
Lire ( Id_Var ) ;
```

Dans cette instruction, l'algorithme attend que l'utilisateur saisisse une valeur à l'aide du clavier ; Ensuite cette valeur sera mise dans la variable **Id_Var**.

Exemple :

```
Algo Lecture
Var X : réel ;
Début

    Lire (X) ; /* X prend la valeur que saisi l'utilisateur à l'aide du clavier */

FIN.
```

40

L'écriture (Affichage)

L'écriture permet d'afficher une valeur à l'écran. Elle est faite à l'aide de l'instruction **Écrire**

```
Écrire ( Message ) ;
```

Message peut être :

- Une Chaîne de Caractères Constante : (Entre Guillemets " ")

```
Écrire ( "Bonjour tout le monde" ) ;
```

- Une Variable :

```
Écrire (X) ; // La valeur de la variable X est affichée
```

- Une Combinaison de Constantes , variables ou expressions séparées par une virgule (,) :

```
Écrire ( "Le valeur de X est " , X ) ;
Écrire ( "Le double de X est " , 2*X ) ;
```

41

Les Entrées / Sorties

Exercice 1:

Écrire un algorithme qui demande à l'utilisateur la longueur et la largeur d'un rectangle, puis calcule et affiche la surface.

```
Algo Surface
Var
Long , Larg , Surf : Réel ;
Début

    Écrire("Donnez la longueur");
    Lire(Long) ;
    Écrire("Donnez la largeur");
    Lire(Larg) ;
    Surf ← Long * Larg ;
    Écrire("La surface est : " , Surf) ;

FIN.
```

42

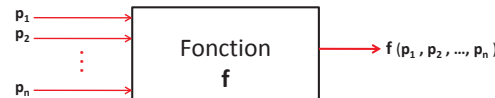
LES FONCTIONS PREDEFINIES

43

Les Fonctions prédéfinies

Qu'est ce qu'une fonction :

Une fonction est un programme (un algorithme) qui calcule une valeur à partir d'un ensemble de paramètres (p_1, p_2, \dots, p_n)



Une fonction est caractérisée par :

- Un identifiant (Un nom).
- Le type de chaque paramètre.
- Le type du résultat qu'elle calcule.

44

Les Fonctions prédéfinies

On dispose d'un ensemble de fonctions prédéfinies que l'on peut utiliser en algorithmique.

Exemples :

- Les Fonctions Mathématiques (**sin**, **cos**, **sqrt** ...)
- Les Fonctions sur les chaînes de caractères (**strlen**, **strcat**...)

45

Les Fonctions prédéfinies

Les Fonctions Mathématique :

1- La Fonction Racine (sqrt)



La fonction sqrt calcule la racine carrée d'un réel. Le résultat est aussi un réel.

Exemple :

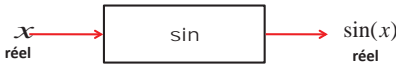
```
Lire (a) ;  
R ← sqrt(a) ; // Les variables R et X doivent être déclarées en réel
```

46

Les Fonctions prédéfinies

Les Fonctions Mathématique :

2- La Fonction sinus



La fonction sin calcule le Sinus d'un angle en **radian** de type réel.
Le résultat est aussi un réel.

Exemple :

```
Lire (a) ;  
S ← sin(a) ;  
Ecrire ("sin(",a,") = ", S);  
// Les variables S et a doivent être déclarées en réel
```

47

Les Fonctions prédéfinies

Les Fonctions Mathématique :

3- La Fonction cosinus



La fonction cos calcule le Cosinus d'un angle en **radian** de type réel.
Le résultat est aussi un réel.

Exemple :

```
Lire (a) ;  
C ← cos(a) ;  
Ecrire ("cos(",a,") = ", C);  
// Les variables C et a doivent être déclarées en réel
```

48

Les Fonctions prédéfinies

Les Fonctions sur les Chaines de Caractères :

La Fonction strlen



La fonction strlen calcule la taille d'une chaine de caractères. C'est-à-dire le nombre de caractères que la chaine contient.

Exemple :

```
Ch ← "Mohammed" ; // Ch est de type chaine de Caractères  
N ← strlen(Ch) ; // N est de type entier (N = 8).
```

49