

# Informatique L1

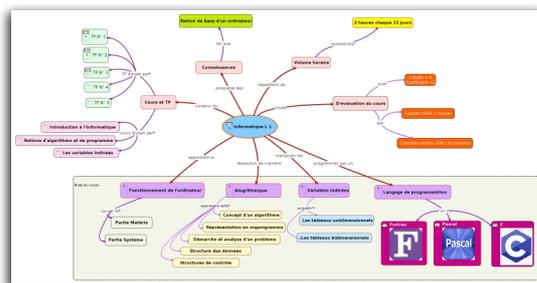
Dr. Yousra Ben Aissa

Université de Mohamed Khider Biskra

Faculté des Sciences Exactes et des Sciences de La  
Nature

Département d' Informatique

E-mail: [yousra.benaissa@univ-biskra.dz](mailto:yousra.benaissa@univ-biskra.dz)



# Table des matières

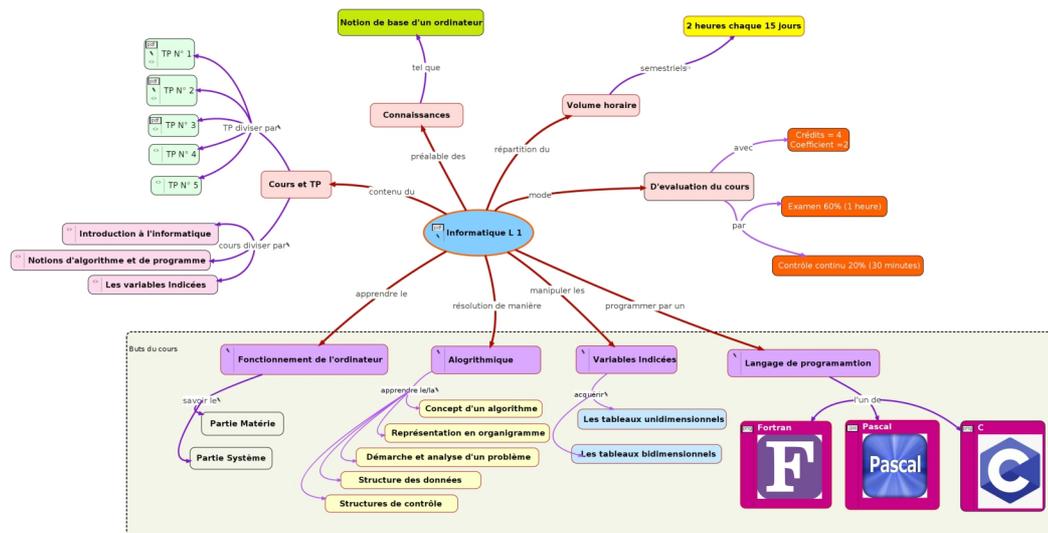
<b>Introduction</b>	<b>3</b>
<b>I - Fiche Contact</b>	<b>4</b>
<b>II - Objectifs Généraux du cours</b>	<b>5</b>
<b>III - Pré-requis / Connaissances préalables nécessaires</b>	<b>6</b>
<b>IV - Exercice : Prétests</b>	<b>7</b>
<b>V - Aide des pré-requis</b>	<b>10</b>
1. Bases de l'informatique.....	10
2. Bases en mathématiques.....	10
3. Bases en anglais.....	10
<b>VI - Chapitre II : Notion d'Algorithme et de Programme</b>	<b>11</b>
1. Concept d'un algorithme.....	11
2. Représentation d'un algorithme.....	12
3. Structures de données .....	13
3.1. Notion d'identificateur.....	13
3.2. Constantes et variables.....	14
3.3. Types de données .....	14
4. Exercice : L'organigramme d'un algorithme .....	14
5. Exercice : Quiz List: Notion d'un Algorithme.....	15
6. Exercice : Structure d'un Algorithme .....	15
<b>Références</b>	<b>17</b>
<b>Bibliographie</b>	<b>18</b>
<b>Webographie</b>	<b>19</b>

# Introduction



Ce module a pour but de développer des compétences dans le domaine d'informatique d'une part et de s'intéresser à les systèmes de codage, l'algorithmique, la programmation d'autre part.

Ce cours est adressé aux étudiants de 1<sup>ère</sup> Année Licence (cycle supérieur) en Science de la matière, Science de la technologie, et Sciences exacte et sciences de la nature de la vie.



[cf. Informatique L1 et L2]

# Fiche Contact

---



 **Complément**

**Établissement** : Université de Mohamed Khider, Biskra

**Faculté** : Sciences exacte et sciences de nature de la vie

**Département** : Informatique

**Public cible** : 1ieme année Licence, spécialité Science de la matière

**Intitulé du cours** : Informatique L1

**Unité d'enseignement méthodologie** : Crédit : 04 | Coefficient : 02

**Durée** : 2 heures chaque 15 jours

**Enseignant** : Dr. Yousra Ben Aissa

**Contact par mail** : yousra.benaissa@univ-biskra.dz

# Objectifs Généraux du cours

---



A l'issue de cet enseignement, l'apprenant sera capable :

1. En terme de Savoir

- Connaître la logique d l'ordinateur.
- Savoir ce qu'est une variable, un tableau de variables, une liste ou une séquence de variables, un branchement conditionnel (if/then/else) et une boucle “for”.
- Savoir ce qu'est une fonction/méthode et un appel récursif.
- Connaître la notion de référence/pointeur, structure/classe.
- Savoir ce qu'est un arbre.
- Savoir ce qu'est un langage de programmation.
- Maîtriser le vocabulaire du langage utilisé.

2. En terme de savoir-faire

- Utiliser un environnement de programmation (Edition, assemblage, compilation etc...).
- Programmer avec un langage évolue (Fortran, Pascal ou C).
- Développer un programme en informatique.

1. En terme de savoir-être

- Construire une algorithmme plus rapide.
- Réduire l'espace utilisé.
- Obtenir une meilleure solution.
- Évaluer la solution obtenu selon l'optimum.

# Pré-requis / Connaissances préalables nécessaires

---



**Fondamental**

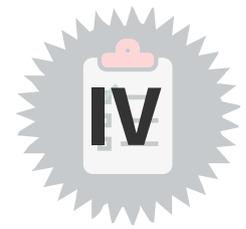
Quels sont donc les prérequis nécessaires ? D'abord, il faut être bien organisé. La programmation, mais aussi l'algorithmique et la plupart des autres domaines de l'informatique, nécessitent de gérer un grand nombre d'objets souvent organisés de manière complexe. Ceci nécessite de l'organisation. Il faut prendre l'habitude et le temps d'organiser ces objets (fichiers, sous-programmes, classes, modules, algorithmes, etc.) afin qu'ils soient faciles à retrouver. Bien sûr, être bien organisé est une chose que l'on peut apprendre. Mais certaines personnes sont naturellement plus organisées que d'autres.

Donc, il est recommandé aux apprenants de connaître :

- Des notions de base d'un ordinateur
- Des notions de base en informatique.
- Les notions de base en mathématiques et en calculs.
- Le vocabulaire technique en informatique et technologie en Anglais
- Comment analyser un problème en étapes et définir les entrées et les sorties

# Exercice : Prétests

---



## Windows :

---

est un

- Système d'entrer et sortie
- Système d'exploitation

## Fenêtres :

---

Opérations de base sur les fenêtres

- Manipuler des fenêtres (les déplacer, les restaurer, etc.)
- Créer et sauvegarder un document texte avec WordPad

## Fichiers et dossiers :

---

identification et manipulations de base

- Déterminer le chemin d'accès complet
- Compresser et décompresser
- Impression : gestion des impressions
- Faire une capture d'écran
- Supprimer et restaurer
- Télécharger des fichiers
- Créer et renommer

## Structures de contrôles :

---

Sélectionner les structures de contrôles

- Si
- Do
- If
- Pour
- Else
- Whether
- Sinon

### Mathématique basique 0:

---

L'ensemble des nombres entiers naturels est noté

- Un nombre dans l'ensemble  $\mathbb{N}$
- Un nombre dans l'ensemble  $\mathbb{Z}$
- Un nombre dans l'ensemble  $\mathbb{D}$
- Un nombre relatif est un nombre entier qui est positif ou négatif
- Un nombre relatif est un nombre entier qui est positif
- Un nombre relatif est un nombre entier qui est négatif
- Un nombre peut s'écrire avec un nombre fini de chiffres après la virgule

### Mathématique basique 1:

---

$$ax+b=0$$

- $x= b/a$
- $x=-b/a$
- $x=a/b$
- $x=-a/b$

### Mathématique basique 2:

---

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

- $\Delta= b^2 + 4ac$
- $\Delta= b^2 - 4ac$

### Anglais :

---

Traduire les instructions suivante en anglais:

Début

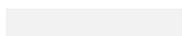
si ( $x=0$ ) alors

$x=2$

sinon

$x=3$

fin.



### Définir les entrées d'un problème :

---

Définir les entrées nécessaires pour calculer la distance entre deux points dans un plan 2D

- Les positions de ces deux points
- Les valeurs de ces deux points
- Les coordonnées de ces deux points

**Définir les sorties d'un problème :**

---

Définir les sorties nécessaires pour calculer la distance entre deux points dans un plan 2D

- Deux points dans un plan 2D
- la distance entre deux points dans un plan 2D
- les distances entre deux points dans un plan 2D

**Analyse un problem en etapes :**

---

exp: calculer la distance entre deux points dans un plan 2D

▯ d'abord ▯, puis ▯ entre les coordonnées de ces deux points, et enfin affichez ▯ en résultat.

# Aide des pré-requis

---



## 1. Bases de l'informatique



---

Les notions de base d'un ordinateur et d'informatique.

[cf. Initiation en informatique]

## 2. Bases en mathématiques



---

Les notions de bases en mathématiques et en calculs.

[cf. Langage mathématiques ensemble de bases]

[cf. Mettre un problème en équation]

## 3. Bases en anglais



---

Vocabulaire technique en Anglais pour l'informatique et technologie

[cf. Vocabulaire Technique Anglais - Mots et expressions de Technologie et Informatique]

# Chapitre II : Notion d'Algorithme et de Programme



L'algorithme désigne la description d'une suite d'actions à effectuer dans un ordre donné pour parvenir à un résultat. La personne ou la machine qui exécute l'algorithme doit savoir effectuer les actions qui y sont énumérés. Les choix des actions élémentaire dépend donc de celui qui doit exécuter l'algorithme.

**Structure d'un algorithme**

Exemple: on veut écrire un algorithme permettant de saluer une personne qui donne son nom.

Spécification du problème:

- Données: nom de la personne
- résultat: afficher 'bonjour' suivi du nom
- principe: saisir le nom, créer le message, puis l'afficher

Algorithme

```
Chaine nom, salutation;  
début  
  écrire "Quel est votre nom?";  
  lire nom;  
  salutation ← "Bonjour" + nom;  
  écrire salutation;  
fin
```

**Calcul de la surface d'un rectangle**

Algorithme CalculSurface :

Var  
Entier : Longueur, Largeur, Surface;

Début  
Écrire (" Donnez la longueur du rectangle: ");  
Lire (Longueur );  
Écrire (" Donnez la largeur du rectangle: ");  
Lire (Largeur );  
Surface ← Longueur \* Largeur ;  
Écrire ("La surface du rectangle est : ");  
Fin.

**Exemple 3**

Ecrire un algorithme qui permet de saisir la valeur du rayon d'un cercle et qui affiche la surface de ce cercle.

Nom : aireDisque  
Rôle : Calculer l'aire d'un disque à partir du rayon de celui-ci  
Entrée :  
Sortie :  
Déclaration : pi : réel  
                  valRayon : Réel  
                  valAire : Réel

Début  
écrire (" Donnez la valeur du rayon ");  
lire (valRayon)  
pi ← 3.14159  
valAire ← pi \* valRayon \* valRayon  
écrire (" L'aire du disque est : ", valAire)

Fin

**Algorithmique et programmation**

Exemple :

Données	Le rayon Pi
Traitement	Surf reçoit: $Pi * R^2$
Résultat	La surface du cercle: Surf

Algorithme Surface\_cercle ;  
Constante Pi = 3,14 ;  
Variable R, Surf : Réel ;  
déclarations  
Début  
Écrire (" Donnez la valeur de rayon: ") ;  
Lire (R) ;  
Surf ← Pi\*R^2 ;  
Écrire (" La surface de cercle est : ",  
Surf) ;  
Fin

Exemple de problèmes résolus par un algorithme

## 1. Concept d'un algorithme

### Algorithme informatique



Le mot Algorithme vient du nom du célèbre mathématicien arabe **Al Khawarizmi\*** (**Abu Ja'far Mohammed Ben Mussa Al-Khwarismi**).

Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné. Pour fonctionner, un algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter.

Un algorithme est une suite ordonnée d'instructions qui indique la démarche à suivre pour résoudre une série de problèmes équivalents.



- préparer une recette de cuisine
- montrer le chemin à un touriste
- programmer un magnétoscope..etc

[cf. Introduction a l'Algorithmique]

## 2. Représentation d'un algorithme

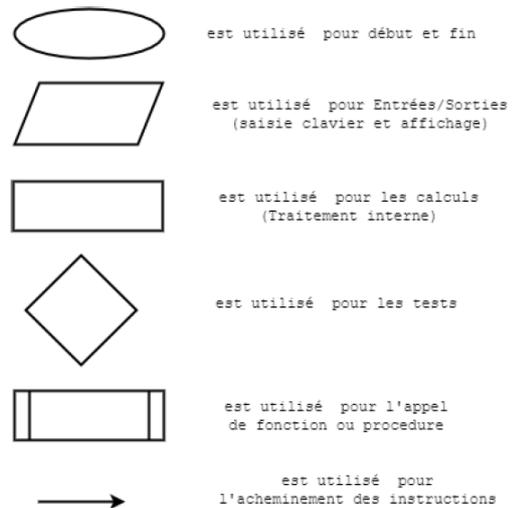
Historiquement, deux façons pour représenter un algorithme \*:

**L'Organigramme** : représentation graphique avec des symboles (carrés, losanges, etc.) qui :

- offre une vue d'ensemble de l'algorithme représentation quasiment abandonnée aujourd'hui Le comportement d'un algorithme
- est déterministe.

**Le pseudo-code** : représentation textuelle avec une série de conventions ressemblant à un langage de programmation. Cette représentation est largement utilisée est plus pratique pour écrire un algorithme.

la figure suivante présente la forme d'un organigramme\*:



*Représentation graphique d'un organigramme*

Un algorithme se décompose en deux grandes parties : d'une part, les déclarations de variables et de constantes, et d'autre part, les instructions à exécuter.\*

```

ALGORITHME<identifiant_algo>
<Partie déclaration>
DEBUT
<Partie Instructions>
FIN
    
```

*Représentation en pseudo code*

**Exemple d'un pseudo code**

Enoncé : additionner deux nombres

**Algorithme 1 Somme**

Var NOMBR1, NOMBR2, SOM : réel;

**Début**

```

1: | Ecrire( 'donner le premier nombre :');
2: | Lire(NOMBR1);
3: | Ecrire( 'donner le deuxième nombre :'); (NOMBR2);
4: | SOM ← NOMBR1 + NOMBR2;
5: | Ecrire('la Somme =', SOM);

```

**Fin**

[cf. Créer un Algorithme]

**3. Structures de données****3.1. Notion d'identificateur**

Un identificateur est une chaîne de caractères contenant uniquement des caractères alphanumériques (alphabétiques de [az] et [AZ] et numérique [09]) et tiré &grave; '\_' (trait souligné), et qui doit commencer soit par une lettre alphabétique ou \_.

- Un identificateur permet d'identifier d'une manière unique un algorithme (ou un programme), une variable, une constante, une procédure ou une fonction.
- Dans un langage de programmation donnée, on a pas le droit d'utiliser les mots réservés (mots clés) du langage comme des identificateurs.



a1 : est un identificateur valide.

a\_1 : est un identificateur valide.

A\_1 : est un identificateur valide.

x12y : est un identificateur valide.

x1 y : est un identificateur non valide (à cause du blanc ou l'espace).

x1-y : est un identificateur non valide (à cause du signe ).

x1\_y : est un identificateur valide.

1xy : est un identificateur non valide (commence un caractères numérique).

### 3.2. Constantes et variables



Les données manipulées par un algorithme (ou un programme) sont soit des constantes ou des variables :

- Constantes : une constante est un objet contenant une valeur qui ne peut jamais être modifiée. Son objectif est d'éviter d'utiliser une valeur d'une manière direct.
- Variables : une variable est un objet contenant une valeur pouvant être modifiée.

Toute les données (variable ou constante) d'un algorithme possède un type de données (domaine de valeurs possibles).

### 3.3. Types de données



Dans l'algorithmique, nous avons cinq types de base :

- Entiers : représente l'ensemble {..., -4,-3,-2,-1,0, 1, 2, 3, 4, ...}.
- Réels : représente les valeurs numériques fractionnels et avec des virgule fixes (ou flottante).
- Caractères : représente tous les caractères imprimable.
- Chaînes de caractères : une séquence d'un ou plusieurs caractères.
- Booléens (logique) : représente les deux valeurs TRUE et FALSE.

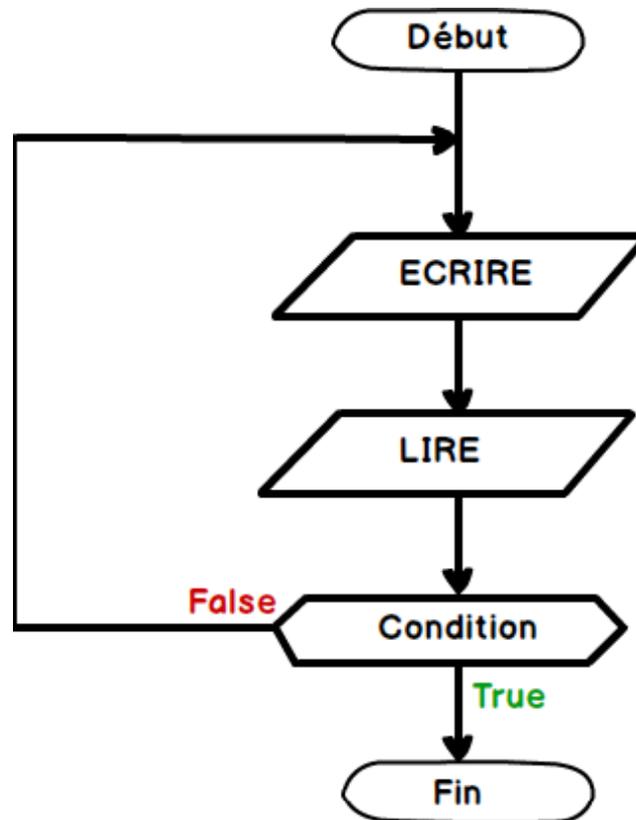


Le tableau suivant montre la correspondance entre les types d'algorithmie et les types du langage de programmation PASCAL.

Algorithme	PASCAL
Entier	Integer
Réel	Real
Booléen	Boolean
Caractère	Char
chaîne	String

## 4. Exercice : L'organigramme d'un algorithme

Quel est l'instruction de lecture dans l'organigramme suivant



## 5. Exercice : Quiz List: Notion d'un Algorithme

### Exercice

Qu'est-ce qu'un algorithme ?

- Un problème de décision
- Une méthode ou un procédé décrit pas à pas
- Un langage de programmation
- Un code numérique

### Exercice

Qu'y a-t-il de commun entre une recette de cuisine et un algorithme ?

- On peut les appliquer comme des théorèmes
- Aucun point commun, une recette n'a rien de mathématique
- Il s'agit d'une suite d'instructions
- Appliquez-les et dans tous les cas, vous obtiendrez un bon résultat

## 6. Exercice : Structure d'un Algorithme

1. `x<- x+1;`
2. `Addition;`
3. `Algorithme`
4. `x: entier;`
5. `Ecrire(x);`
6. `Ecrire("msg");`
7. `Lire(x);`
8. `Début`
9. `Var`
10. `Fin.`

*Réponse :*

# Références

---



*L'invention des algorithmes et la naissance de l'algèbre* <https://www.pourlascience.fr/sd/histoire-sciences/l-invention-des-algorithmes-et-la-naissance-de-l-algebre-4842.php>

*structure d'un algorithme* [https://www.technologuepro.com/informatique/chap1\\_algorithme.htm](https://www.technologuepro.com/informatique/chap1_algorithme.htm)

# Bibliographie

---



[https://info.uqam.ca/~privat/INF2170/notes\\_de\\_cours/ndc02-structure\\_fonctionnement.pdf](https://info.uqam.ca/~privat/INF2170/notes_de_cours/ndc02-structure_fonctionnement.pdf)

<https://www.technologuepro.com/cours-systemes-logiques-3/chapitre-1-1-systeme-de-numeration-et-codage-des-informations.pdf>

# Webographie

---



<https://www.larousse.fr/dictionnaires/francais/informatique/42996>

<https://www.calculquebec.ca/vitrine/histoire/>

<https://rascolsin.fr/psti2d/algo/01-Algo-Orgas/#2-lorganigramme>

<https://rascolsin.fr/psti2d/algo/01-Algo-Orgas/#1-lalgorithme>