



**Algorithmique et Structure de données statiques**  
**Travaux dirigés**  
**Première Partie**  
**Variables, instructions de base, trace d'un algorithme**

**Questions de cours**

1. Définir la notion d'algorithme
2. Définir la notion de variable
3. Quelles sont les caractéristiques d'une variable ?
4. Quels sont les différentes catégories de variables qui existent ?
5. Citer les différents types scalaires prédéfinis.
6. Une variable peut prendre combien de valeur à la fois
7. Quelle est la différence entre une constante et une variable ?
8. Citer et expliquer les différentes instructions de base
9. Donner un exemple d'application de chaque instruction de base.
10. Qu'est-ce qu'une procédure ?
11. Qu'est-ce qu'une fonction ?
12. Quelles sont les différences entre une procédure et une fonction ?
13. Donner les différentes parties d'un algorithme
14. Citer les différentes étapes de construction d'un algorithme
15. Donner la structure d'un algorithme
16. Définir et donner l'intérêt d'un type Enregistrement.
17. Ecrire un algorithme résolvant un problème de votre choix.

## Manipulation des variables et affectation

Exercice 1	Exercice 2	Exercice 3
Quelles seront les valeurs des variables $a$ et $b$ à la fin de l'algorithme <i>Algo1</i> ? Faites une trace !	Quel est le résultat de la fonction <i>Algo2</i> (3,5) ? Faites une trace !	Écrire un algorithme avec les instructions suivantes, et faites-en la trace.
<b>Procédure <i>Algo1</i></b> Variables locales : entiers $a$ et $b$ Début $a \leftarrow 1$ $b \leftarrow a+3$ $a \leftarrow 3$ <b>FinProcédure <i>Algo1</i></b>	<b>Fonction <i>Algo2</i> est de type entier</b> paramètres en entrée : entiers $a$ et $b$ paramètres en sortie: entier $c$ Début $c \leftarrow a+b$ $a \leftarrow 2$ $c \leftarrow b-a$ renvoyer $c$ <b>FinFonction <i>algo2</i></b>	1. J'initialise $a$ à 8 2. Je stocke $a+1$ dans $b$ 3. J'ajoute 1 à $a$ 4. Je multiplie $a$ par 2 5. Je retranche 5 à $b$ 6. Je multiplie $b$ par lui-même 7. J'ajoute 3 à 2

## Exercice 4

<b>Fonction <i>mystere</i> est de type entier</b> <b>paramètres en entrée :</b> entiers $u$ et $v$ <b>paramètres en sortie:</b> entier $r$ <b>Début</b> Tant que $v > 0$ faire $r = u \bmod v$ ; $u = v$ ; $v = r$ ; Fin Tant que $r = u$ ; renvoyer $r$ ; <b>FinFonction <i>mystere</i></b>
---

1. Faire la trace de l'appel *mystere*(84, 36)
2. Que fait la fonction *mystere*.

**Exercice 5 :** Écrire un algorithme qui convertit une température entrée au clavier exprimée en degrés Fahrenheit en degrés Celsius avec la formule suivante :  $C = 0.55556 \times (F - 32)$ . F est une température en degrés Fahrenheit et C la température correspondante en degrés Celsius.

**Exercice 6 :** Écrire un algorithme qui demande à l'utilisateur de saisir les trois coefficients (a, b et c) d'une équation du second degré et qui affiche les solutions de l'équation.

# Algorithmique et Structure de données statiques

## Travaux dirigés

### Deuxième partie

### Structure de contrôle

#### Exercice 1

Ecrire un algorithme qui permet de convertir le franc CFA en Euro en ou Dollar en fonction du choix de l'utilisateur entré au clavier.

Pour information ;

1 **euro** = 655,955 F CFA

1 **Dollar** = 500 F CFA

#### Exercice 2

Ecrire un algorithme qui demande le nombre de crédits obtenus par un étudiant dans un semestre et qui affiche s'il a validé ou non le semestre. Un semestre est validé si l'étudiant a obtenu 30 crédits.

#### Exercice 3

Écrire un algorithme demandant à l'utilisateur de saisir la valeur d'une variable n et qui affiche la table de multiplication de n.

#### Exercice 4

Écrire un algorithme qui permet de calculer la racine d'un entier n saisi au clavier avec la formule

de récurrence suivante :

$$\begin{cases} u_0 = \frac{n}{2} \\ u_i = \frac{1}{2} \left( u_{i-1} + \frac{n}{u_{i-1}} \right) \end{cases}$$

#### Exercice 5 : la suite de Fibonacci

Ecrire une fonction qui permet de calculer la suite de *Fibonacci* d'un entier n en entrée et de retourner le résultat.

En mathématique, la suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence par les termes 0 et 1. Les termes de cette suite sont appelés *nombre de Fibonacci*.

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

Par exemple  $F_2 = F_1 + F_0 = 1 + 1 = 2$

# Algorithmique et Structure de données statiques

## Travaux dirigés

### Troisième partie

### Tableau

**Exercice 1 :** Écrire un algorithme qui permet de ranger les coordonnées d'un vecteur dans un tableau de trois éléments. L'algorithme affiche ensuite les coordonnées du vecteur.

**Exercice 2 :** Écrire un algorithme qui affiche le nombre de valeurs paires et le nombre de valeurs impaires dans un tableau d'entiers de dix éléments.

**Exercice 3 :** Écrire un algorithme qui détermine la plus grande valeur et la plus petite valeur d'un tableau d'entiers. Seul le premier maximum (minimum) est considéré.

#### Exercice 4 : Traitement des notes des étudiants

On a à disposition un tableau  $t$  d'entiers correspondant aux notes d'un étudiant : [15,9,10,17,8,14].

1. Écrire un algorithme **cherche** qui prend en entrée un tableau d'entiers  $t$  ainsi qu'un entier  $a$  et retourne le booléen VRAI si l'entier  $a$  est dans le tableau  $t$ , et FAUX sinon.
2. Quel est le résultat de l'appel suivant **cherche**([15,9,10,17,8,14], 18) ?
3. Écrire un algorithme **somme** qui prend en entrée un tableau d'entiers  $t$  et qui renvoie la somme des entiers contenus dans  $t$ .
4. Quel est le résultat de l'appel suivant **somme**([15,9,10,17,8,14]) ?

#### Exercice 5

Un palindrome est un tableau de caractères qui se lit indifféremment dans les deux sens (par exemple : radar, la mariée ira mal).

1. Écrire une fonction qui retourne 1 si un tableau de caractères passé en argument est un palindrome, 0 sinon.
2. Écrire une fonction qui vérifie si une chaîne de caractère passé en argument est un palindrome. On traitera une chaîne de caractères comme un tableau de caractères.

#### Exercice 6

Écrire un algorithme qui permet de faire la somme de deux polynômes de même degré passés en entrée et de retourner le résultat. On suppose que les polynômes sont définis sous forme de tableau à une dimension. Les éléments du tableau sont des coefficients et les indices du tableau correspondent aux degrés des coefficients. Le degré d'un polynôme est la longueur du tableau diminuée de 1.

par exemple :

$P(x) = 2x^3 + 5x^2 + 9x - 1$  est représenté par le tableau suivant :

-1	9	5	2
----	---	---	---

$P_1(x) = 10x^5 + 3$  est représenté par le tableau suivant :

3	0	0	0	0	10
---	---	---	---	---	----

### Exercice 7

Ecrire un algorithme qui permet de faire la soustraction de deux polynômes en entrée et de retourner le résultat obtenu.

### Exercice 8

Ecrire un algorithme qui permet de faire la multiplication de deux polynômes en entrée et de retourner le résultat.

### Exercice 9: Tableau à deux dimensions

Le but de cet exercice est de manipuler des tableaux de tableaux de booléens qui représentent des images en noir et blanc. En effet, on va coder une image en noir et blanc de la manière suivante : chaque pixel noir correspond au booléen FAUX (pixel éteint) et chaque pixel blanc au booléen VRAI (pixel allumé). Un tableau de booléens correspond donc à une colonne de pixels, et un tableau de tableaux de booléens correspond à une image. Par exemple, l'image ci-dessous correspond au tableau

[[VRAI,VRAI,VRAI],  
[VRAI,VRAI,FAUX],  
[FAUX,FAUX,VRAI]].


# Algorithmique et Structure de données statiques

## Travaux dirigés

### Quatrième partie

### Les enregistrements

**Exercice 1 :** Écrire un algorithme qui définit un type de données **POINT** de type enregistrement ayant trois champs de type réel pour représenter les coordonnées d'un point. Utiliser ce nouveau type pour calculer la distance à l'origine d'une variable de type enregistrement **POINT**, et la distance entre deux variables de type enregistrement **POINT**.

**Exercice 2 :** Écrire un algorithme qui définit un type de données **COMPLEXE** de type enregistrement ayant deux champs de type réel pour représenter un nombre complexe. Utiliser ce nouveau type pour réaliser des opérations sur des variables de type **COMPLEXE** : addition, multiplication, calcul de la norme, calcul de l'argument.

#### **Exercice 3 :**

On souhaite implémenter un dictionnaire français simplifié. Un dictionnaire est un ensemble de mots. Chaque mot a une définition. Les mots sont classés par ordre alphabétique. A partir d'une lettre, le dictionnaire doit permettre de retourner tous les mots associés. En plus, le dictionnaire doit permettre de retrouver et d'insérer la définition d'un mot.

Nous avons les structure de données suivantes :

**Mot** : il est caractérisé par le terme (par exemple "Maison", "Elephant") qui le désigne et sa définition (une chaîne de caractères).

**SousDictionnaire** : il est caractérisé par une lettre d'alphabet ("**b**" par exemple) et un tableau de **Mot** qui correspond à la liste des mots qui commencent par cette lettre.

**Dictionnaire** : il est caractérisé par un tableau de **SousDictionnaire**.

1. Définir la structure de données permettant de représenter un **Mot** ;
2. Définir la structure de données permettant de représenter un **SousDictionnaire** ;
3. Définir la structure de données permettant de représenter un **Dictionnaire**.
4. Proposer une fonction **saisirMot** qui permet à l'utilisateur de saisir un terme et sa définition. Elle retourne un résultat de type **Mot**.
5. Proposer une fonction **saisirSousDictionnaire** qui permet à l'utilisateur de créer un sous dictionnaire. Cette fonction appellera la fonction **SaisirMot**.
6. Proposer un algorithme **chercherMot** qui permet de vérifier si un mot existe dans le dictionnaire. Il retourne la valeur 0 si le mot existe sinon 1.
7. Proposer une procédure **ajouterMot** qui permet d'ajouter un mot dans un sous-dictionnaire. La procédure prend en argument un sous-dictionnaire, le mot à ajouter et la première lettre du mot.