MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

Formation au Ministère de l'Agriculture

Les fondamentaux de la programmation

MINISTÈRE
DE L'AGRICULTURE
DE LA SOUVERAINETÉ
ALIMENTAIRE ET DE LA FORÊT
Liberté
Égalité
Fraternité

Sommaire

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

0.1 Avant-propos

Ce diaporama de formation a été rédigé dans le but d'être le support visuel des formations dispensées au MASA.

Cette formation s'adresse à des agents souhaitant découvrir l'écriture de syntaxe avec R sans toutefois avoir déjà écrit des programmes dans d'autres langages.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

0.2 Avant-propos

La formation est donnée en distancielle sur une durée **d'une journée**.

Objectifs par cette formation

Ce support permet de :

- Connaître les principes fondamentaux de la programmation
- Avoir les pré-requis nécessaires à la formation « Initiation à la programmation R »

Pour information, les thèmes abordés sont:

- Algorithmes et programmes
- Les variables : nommage et typage
- Les différentes instructions : affectation, lecture, écriture ….
- Les tests de conditionnalité
- Les fonctions : utilité, arguments, résultats

~~Ils sont orientés pour être utile aux agents du SSM Agriculture et se concentrent sur une utilisation de R via RStudio qui est mise à disposition des agents sur la~~ plateforme interne Cerise basée sur RStudio Workbench.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1 Algorithmique et Programmation

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1.1 Etapes de développement d'un logiciel ou programme

ANALYSER Comprendre les besoins du client.

- Dossier d'analyse
- Cahier des charges

CONCEVOIR Identifier tout ce qui va être utile pour répondre au besoin du client. => Algorithmique

- Documents de spécifications détaillées

REALISER Implémenter la solution choisie pour répondre au besoin du client. => Programmation

- Programmes prêts à exécuter

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1.2 Définitions

- Représentatif d'une logique
- Indépendant de tout langage de programmation
- Inexploitable en l'état par un ordinateur
- Compréhensible par tous
- Suite d'instructions ou d'étapes
- Plusieurs représentations : schémas, texte (pseudo-code)

- Traduction d'un algorithme dans un langage de programmation spécifique (lignes de code)
- Exécutable par le système

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1.3 Analogie avec la littérature dans le cadre d'un livre traduit en plusieurs langues

- L'écrivain réalise l'algorithme (imagination du récit, choix des personnages, découpage en chapitres….) et écrit le programme dans une langue donnée
- Les traducteurs transposent l'œuvre de l'écrivain dans d'autres langues en essayant d'être le plus fidèles possibles à l'histoire et aux émotions qu'il a voulu transmettre.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1.4 Structure d'un algorithme en pseudo-code

- Choix du nom de l'algorithme
- Déclaration des variables manipulées
- Initialisation des variables et liste des instructions

```
ALGORITHME < Nom >

VARIABLES
        < Liste des variables >

DEBUT
        < Liste des instructions >

FIN
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

1.5 Plusieurs variantes correctes du bloc d'instructions existent.

- Le choix des instructions tient compte de plusieurs facteurs :
  - optimisation,
  - lisibilité et compréhension,
  - possibilité de traduction dans plusieurs langages différents ….
- Le pool d'instructions disponibles peut être lié au langage de programmation mais aussi à la communauté de développeurs (librairies mises à disposition….).

Exercices 1 à 3 (manipulation des instructions)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

2 Variables et Types

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.1 Définitions

- Une **variable** est un symbole qui porte un nom et qui contient une valeur d'un type donné.

Si la valeur de la variable n'est pas modifiable dans le temps, le terme de **constante** peut être utilisé.

- **Déclaration :** indiquer l'existence d'une variable et préciser son type

```
<nom_variable> : <nom_type>
```

- **Initialisation :** donner une valeur initiale à une variable
- **Affectation/assignation** : donner une valeur à une variable à tout moment. La nouvelle valeur affectée à une variable écrase l'ancienne valeur.

```
<nom_variable> <- <valeur>
```

Exercices 4 (manipulation d'une variable)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.2 Principaux types

Type	Utilité	Exemple
Booléen	Représente un état binaire, vrai ou faux	`Vrai`
Numérique	Représente un nombre quelconque	`12345.6789`
Caractère	Représente un caractère unique (comme une touche du clavier)	`'c'`
Chaîne de caractères	Représente un texte	`"contenu de la chaîne"`

Il est possible de ventiler la catégorie « **Numérique** » en 2 catégories distinctes :

- Les entiers
- Les réels

La famille des « **dates** » peut également être considérée comme un type de base pour les variables.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.3 Opérations possibles

Pour les **numériques** : toutes les **opérations mathématiques** classiques

- **+** : addition
- **-** : soustraction
- ***** : multiplication
- / : division

Pour les **chaînes de caractères** :

- **+** : concaténation (assembler des caractères à la suite des autres)

Chaque langage de programmation décline ces types de base en plus ou moins de types de données propres. Le type choisi au moment de l'écriture du programme aura un impact sur l'allocation mémoire qui sera faite pour le stockage de la variable mais aussi sur la liste des fonctions utilisables.

Exercices 5 (manipulation de variables de types différents)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.4 La valeur `NULL` (1/2)

Le mot clé `NULL` est utilisé pour indiquer que la valeur d'une variable est inconnue.

Remarque :

- Pour des numériques : `NULL` est différent de 0
- Pour des chaines de caractères : `NULL` est différent de « »
- Pour des booléens : `NULL` est différent de FAUX

Des opérations simples peuvent entraîner des erreurs (plantages) si des variables ne sont pas valorisées.

Exemple : addition de 2 numériques dont l'un a pour valeur `NULL`.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.5 La valeur `NULL` (2/2)

Pour éviter ce type d'erreur :

- Initialiser le plus souvent possible les variables en début de programme
- Si le langage le permet, utiliser la notion de valeur par défaut lors de la déclaration des variables
- Si le langage le permet, utiliser une fonction permettant de remplacer toutes les valeurs NULL par une valeur passée en paramètre

Dans certains cas, une valeur vide peut être significative. Un mot clé spécifique au langage est utilisé.

Dans ce cas, la notion de « non applicable » ou « valeur indéfinie » est souvent employée.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.6 Les tableaux

Un **tableau** est une variable qui peut contenir plusieurs valeurs simultanément.

Un tableau se présente comme une structure organisée sous forme de cellules ou cases qui peuvent être identifiées par un indice ou index.

Selon les langages de programmation, certaines règles peuvent différer :

- Un tableau peut ne contenir que des variables du même type
- La valeur de l'indice débute à 0 ou à 1

Un tableau peut avoir 1 ou plusieurs dimensions.

- 1 dimension : vecteur ou liste
- Plusieurs dimensions : matrice ou tableau multi-dimensionnel

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

2.7 Les tableaux

Les tableaux les plus fréquemment exploités au MASA sont des **tables de données**.

Prénom	Sexe	Taille
Pierre	H	185
Anne	F	167
Marie	F	171
Charles	H	175
Caractères	*Caractères*	*Numériques*

Exemple d'un support de cours « Initiation à R »
Tableau TAB_IND

- = <nom_tableau>[num_ligne,num_colonne] = TAB_IND[3,2]
- = <nom_tableau>$<nom_colonne>[num_ligne] = TAB_IND$Sexe[3]

Une **colonne** ne peut contenir que des valeurs de même type.

`Colonne = variable = vecteur`

Toutes les colonnes possèdent le même nombre de valeurs. Un nom peut être attribué à chaque colonne. Ce nom de colonne peut être utilisé pour accéder à la valeur de la cellule.

Une **ligne** contient les valeurs observées pour chaque variable pour un élément donné. Ligne = observation = occurrence = élément

~~Les **indices ou index** de lignes et de colonnes débutent à 1.~~

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3 Les Tests

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.1 La condition - “SI”

SI < condition > = VRAI
ALORS
< liste d'instructions >
SINON
< liste d'instructions >
FIN SI

- La condition est une expression booléenne qui ne peut avoir que 2 valeurs possibles : VRAI ou FAUX.
- Le bloc d'instructions SINON est facultatif.
- Plusieurs conditions SI peuvent être imbriquées.

Analogie avec Excel :

SI(test_logique;valeur_si_vrai;valeur_si_faux)
Vérifie si la condition est respectée et renvoie une valeur si le résultat d'une condition que vous avez spécifiée est VRAI, et une autre valeur si le résultat est FAUX.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

3.2 La condition - “SI”

La condition exprimée dans un bloc SI peut correspondre à toute expression pouvant être valorisée par **VRAI** ou **FAUX**.

Cette expression peut être unitaire ou combinée par des opérateurs logiques (ET et OU). Les parenthèses sont utilisées pour encadrer les différentes conditions et gérer les priorités.

Exemples de conditions courantes :

- est égal à , différent de, supérieur, inférieur ….
- contient, commence par …
- est vide, existe …

Exemples de conditions combinées :

- Condition A ou Condition B
- (Condition A et Condition B) ou Condition C

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.3 Exemples

```
ZoneOutreMer <- FAUX
SI CodeDepartement possède 3 caractères
ALORS
    ZoneOutreMer <- VRAI
FIN SI
```

```
SI MoyenneGenerale >= 10
ALORS
    ResultatExamen <- « Reçu »
SINON
    ResultatExamen <- « Ajourné »
FIN SI
```

```
SI MoyenneGenerale >= 10
ALORS
    ResultatExamen <- « Reçu »
    Mention <- « »
    SI MoyenneGenerale >= 16
    ALORS
        Mention <- « Très Bien »
    SINON
        SI MoyenneGenerale >= 14
        ALORS
            Mention <- « Bien »
        SINON
            SI MoyenneGenerale >= 12
            ALORS
                Mention <- « Assez Bien »
            FIN SI
        FIN SI
    FIN SI
SINON
    ResultatExamen <- « Ajourné »
    Mention <- «  »
FIN SI
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.4 Les opérateurs logiques

La valeur d'une condition combinée dépend des valeurs de chaque condition unitaire utilisée, des opérateurs logiques et des priorités données par les parenthèses.

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.5 L'opérateur "OU"

« Pour que la condition combinée soit vraie, au moins une des 2 conditions doit être vraie ».

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

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.6 L'opérateur "ET"

« Pour que la condition combinée soit vraie, les 2 conditions doivent être vraies »

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

~~En l'absence de parenthèses, l'opérateur **ET** est prioritaire sur l'opérateur **OU**.~~

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.7 L'opérateur "OU EXCLUSIF"

« Pour que la condition combinée soit vraie, au moins une des 2 conditions mais pas les 2 doit être vraie ».

Condition A	Condition B	A OU exclusif B
VRAI	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

Illustration fonctionnelle :

Cas d'un menu proposant une formule « Plat + entrée ou dessert » à 20€

- Si le client prend un plat et une entrée, il peut bénéficier de la formule
- Si le client prend un plat et un dessert, il peut bénéficier de la formule
- Si le client prend un plat, une entrée et un dessert, il ne peut pas bénéficier de la formule.

Exercices 6 (valorisation d'une condition)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.8 Les Tests - "SELON"

SELON < valeur de la variable testée > **FAIRE**

CAS < liste 1 de valeurs > : < liste d'instructions >
CAS < liste 2 de valeurs > : < liste d'instructions > ….

CAS < liste N de valeurs > : < liste d'instructions >

AUTREMENT : < liste d'instructions >

FIN SELON

La liste de valeurs peut contenir une seule valeur, une suite de valeurs séparées par des virgules et/ou un intervalle de valeurs.

Exercices 7 et 8 (utilisation du test SELON)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

3.9 Exemples

```
SI MoyenneGenerale >= 10
ALORS
    ResultatExamen <- « Reçu »
    Mention <- « »
    SI MoyenneGenerale >= 16
    ALORS
          Mention <- « Très Bien »
    SINON
          SI MoyenneGenerale >= 14
          ALORS
                Mention <- « Bien »
          SINON
                SI MoyenneGenerale >= 12
                ALORS
                      Mention <- « Assez Bien »
                FIN SI
          FIN SI
     FIN SI
SINON
    ResultatExamen <- « Ajourné »
    Mention <- «  »
FIN SI
```

```
Mention <- «  »

SI MoyenneGenerale < 10
ALORS
        ResultatExamen <- « Ajourné »
SINON
        ResultatExamen <- « Reçu »
        SELON MoyenneGenerale FAIRE
                CAS [12-14[ : Mention <- « Assez Bien »
                CAS [14-16[ : Mention <- « Bien »
                CAS >= 16 : Mention <- « Très Bien »
        FIN SELON
FIN SI
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

4 Les Boucles

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

4.1 Définition

Une **boucle** permet de répéter plusieurs fois un bloc d'instructions.

3 types de boucles coexistent :

- Boucle « **POUR** » : elle est utilisée lorsque le nombre de répétitions est connu à l'avance
- Boucle « **TANT QUE** » : elle est utilisée lorsque le nombre de répétitions n'est pas connu à l'avance et permet de répéter l'exécution du bloc d'instructions tant que la condition de bouclage est respectée
- Boucle « **REPETER** » ou « **FAIRE TANT QUE** » : variante de la boucle « TANT QUE » où le bloc d'instructions est exécuté avant le test de la bouclage.

Il est obligatoire de s'assurer que la condition de bouclage deviendra FAUSSE à un moment donné pour provoquer l'arrêt de l'exécution du bloc d'instructions. Dans le cas contraire, le programme ne s'arrêtera jamais (boucle infinie).

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

4.2 Les boucles POUR

POUR `<variable>` **ALLANT DE** `<debut>` **A** `<fin>`

FAIRE < liste d'instructions >

FIN POUR

Illustration :

Boucle permettant d'afficher le libellé des 12 mois de l'année.

```
VARIABLES :
    indice_boucle : ENTIER
    libelle_mois : CHAINE de CARACTERES
DEBUT
  POUR indice_boucle ALLANT DE 1 A 12
  FAIRE
      libelle_mois <- donne_libelle_du_mois(indice_boucle)
      Afficher libelle_mois
  FIN POUR
FIN
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

4.3 Les boucles TANT QUE

TANT QUE = VRAI

FAIRE < liste d'instructions >

FIN TANT QUE

Illustration :

Boucle permettant d'afficher le libellé des 12 mois de l'année.

```
VARIABLES :
    indice_boucle : ENTIER
    libelle_mois : CHAINE de CARACTERES
DEBUT
  Indice_boucle <- 1
  TANT QUE indice_boucle <= 12
  FAIRE
      libelle_mois <- donne_libelle_du_mois(indice_boucle)
      Afficher libelle_mois
      indice_boucle <- indice_boucle + 1
  FIN TANT QUE
FIN
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

4.4 Les boucles REPETER

REPETER

< liste d'instructions >

JUSQU'À = VRAI

FIN REPETER

Illustration :

Boucle permettant d'afficher le libellé des 12 mois de l'année.

```
VARIABLES :
    indice_boucle : ENTIER
    libelle_mois : CHAINE de CARACTERES
DEBUT
  Indice_boucle <- 1
  REPETER
      libelle_mois <- donne_libelle_du_mois(indice_boucle)
      Afficher libelle_mois
      indice_boucle <- indice_boucle + 1
      JUSQU'À indice_boucle = 13
  FIN REPETER
FIN
```

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

5 Les Fonctions

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.1 Définition (1/2)

Une **fonction** correspond à un regroupement d'instructions dans le but de réaliser une action précise et qui a vocation à être réutilisé.

Chaque langage de programmation propose un pool de fonctions de base plus ou moins riche qui peuvent être regroupées ou non dans des librairies.

Tout développeur peut utiliser des fonctions déjà écrites ou créer des fonctions personnalisées pour des traitements répondant à ses besoins.

<u>Une fonction :</u>

- Possède un nom
- Peut nécessiter des données en entrée (paramètres)
- Peut fournir une valeur en retour (paramètre de sortie ou résultat)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

5.2 Définition (2/2)

Toutes ces informations sont mentionnées dans la 1er ligne du script qui se nomme la **signature**.

Plusieurs fonctions portant le même nom peuvent coexister. Dans la majorité des cas, elles possèdent des signatures différentes.

Exemple :

La fonction **isNull(**`<variable>`**)** qui retourne un booléen. Cette fonction a été implémentée dans de nombreux langages pour des paramètres d'entrée de plusieurs types (entier, chaines de caractères ….)

MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE
Liberté
Égalité
Fraternité

5.3 Syntaxe

FONCTION `<type du résultat> <nom de la fonction> (<type du param1>` <nom du param1, ...)

< liste d'instructions >

RETOURNER `<valeur du résultat>`
FIN

Une fonction est dite **récursive** lorsqu'elle s'appelle elle-même. Dans ce cas, son code doit forcément comprendre une condition d'arrêt.

L'utilisation de fonctions permet souvent de décomposer un traitement complexe en plusieurs traitements plus simples. La compréhension et la maintenance des programmes sont alors facilitées.

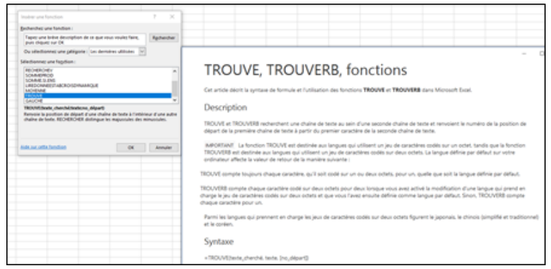
Toute fonction doit être **documentée** pour permettre sa réutilisation ultérieure. En général, l'espace de développement donne la possibilité d'accéder à la description de la fonction et éventuellement à des exemples d'utilisation. Les sites web de communautés de développeurs sont aussi très riches en informations.

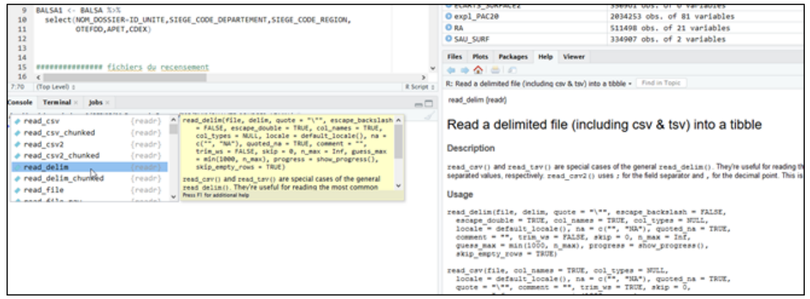
MINISTÈRE
DE L'AGRICULTURE
ET DE LA SOUVERAINETÉ
ALIMENTAIRE

Liberté
Égalité
Fraternité

5.4 Usage

Avant de se lancer dans l'écriture d'une fonction, il faut au préalable rechercher si elle n'existe pas déjà.





Exercice 9-10-11 (tableaux, boucles, tests et fonctions)