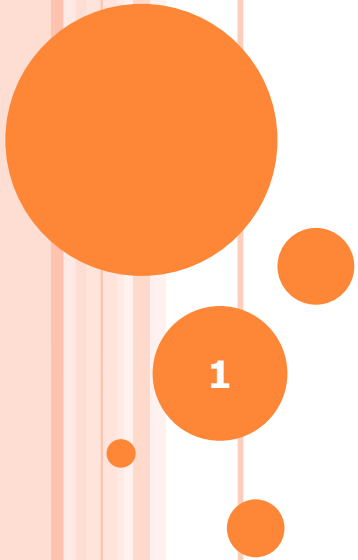
Licence STS

Université Claude Bernard Lyon I

LIFAP1 : ALGORITHMIQUE ET PROGRAMMATION IMPÉRATIVE, INITIATION



COURS 5 : Les Tableaux 2D

PLAN DE LA SÉANCE

- Apprendre à manipuler des tableaux
  - 2 dimensions
  - Multi-dimensions
- Application au cas particulier des ensembles

TABLEAU À 2 DIMENSIONS

- Déclaration :

T : tableau [10] [5] d'entiers

- T sera un tableau
  de 10 lignes
  et 5 colonnes
- Accès :
  - T[i-1][j-1]
  - désigne la case
    à la $i^{ème}$ ligne
    et $j^{ème}$ colonne

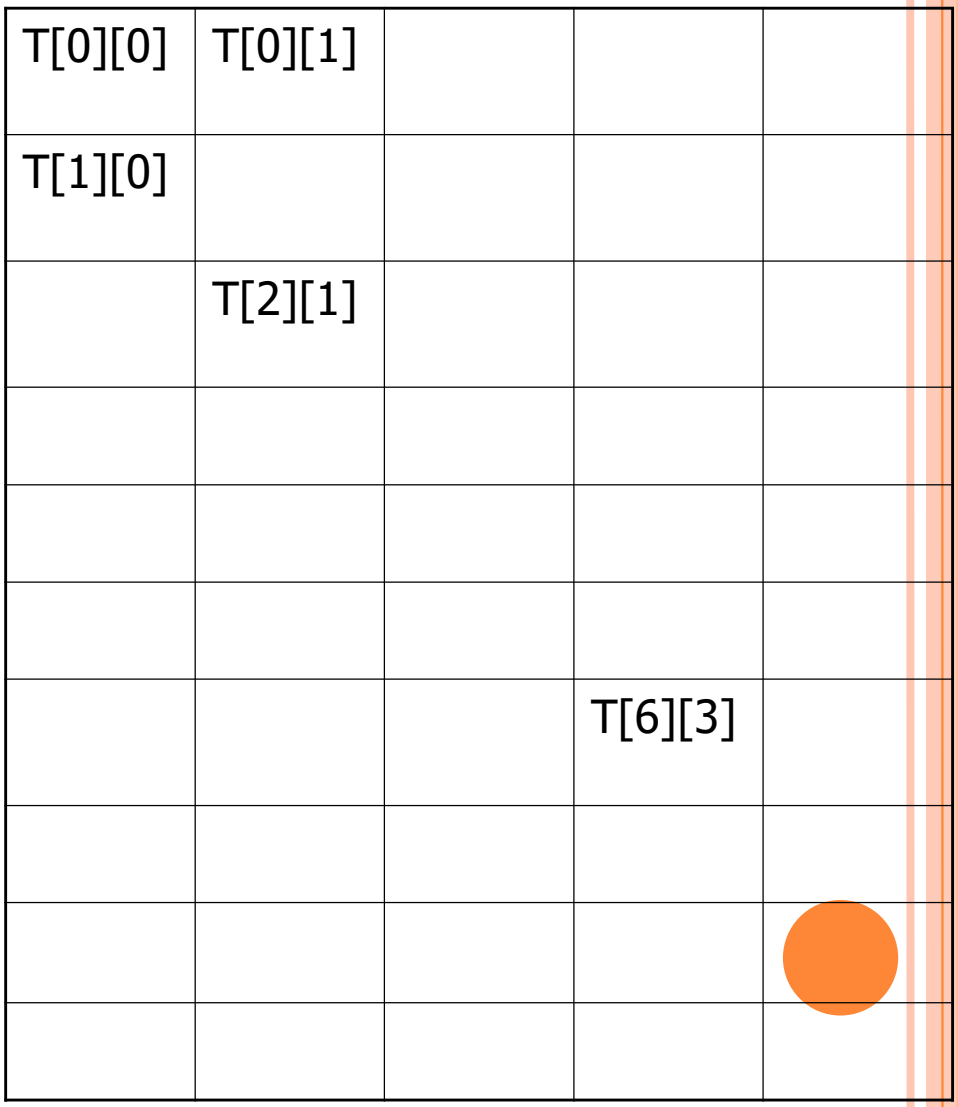


TABLEAU À 2 DIMENSIONS : UTILITÉ

- Modélisation de la notion mathématique de matrice
- Modélisation d'une grille :
  - Bataille navale
  - Tétris
- Modéliser une surface ou un plan

INITIALISATION

procédure initialisationA0 (T : tableau[10][10] de entier)

préconditions : aucune

donnée/résultat : T

description : met des 0 dans toutes les cases du tableau 2D

variable locale : i : entier, j : entier

début

Pour i **allant de** 0 à 9 **par pas de** 1 **faire**

Pour j **allant de** 0 à 9 **pas de** 1 **faire**

T[i][j] ← 0

Fin Pour

Fin Pour

fin

LA MATRICE IDENTITÉ

- Matrice carrée : tableau de taille n*n
- Des 0 partout sauf sur la diagonale : si i=j alors on met un 1
- Algorithme de remplissage
  - On initialise dans un premier temps avec que des 0 (initialisationA0)
  - On met les 1 sur la diagonale T[i][i]

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

LA MATRICE IDENTITÉ : “POUR”

procédure identité (T : tableau[10][10] de entier)
préconditions : aucune
donnée/résultat : T
description : met des 1 sur la diagonale du tableau
variable locale : i : entier
début
initialisationA0(T)
Pour i **allant de** 0 à 9 **par pas de** 1 **faire**
T[i][i] ← 1
Fin Pour
Fin

REMARQUES, EXTENSION ND

- Comme pour les tableaux 1 dimension, les nombres de lignes et de colonnes effectivement utilisées peuvent être passés en paramètres :
  - taille dans chaque dimension
  - indices début et fin dans chaque dimension (bloc)
  - ➢ Utilisation partielle de la matrice
- Nombre de dimensions aussi grand que l'on veut :
  - T3 : tableau [N][M][O] de truc
    - à 3 dimensions
  - Limitation dues :
    - Représentation graphique et "visuelle" difficile pour programmeur
    - Manipulation des indices

STRUCTURE ABSTRAITE : L'ENSEMBLE

- Application directe des tableaux
- Objet mathématique

- Restriction à un ensemble fini
  - Chaque élément est unique
  - Une valeur appartient ou n'appartient pas à un ensemble
  - Opérations sur les ensembles :
    - Union / intersection / différence
  - Ordre partiel : relation d'inclusion

ENSEMBLE VIA L'UTILISATION D'UN TABLEAU

- Déclaration du tableau :
  - Dimension = cardinalité maximale de l'ensemble

- Si toutes les positions du tableau ne sont pas significatives, il faut mémoriser celles qui contiennent des données valides :
  - généralement placées en début de tableau
  - une variable indique le nombre de positions valides à partir du premier indice
    (n éléments occupent les indices compris entre 0 et n-1)

TABLEAU DES 10 PREMIÈRES VALEURS DE LA FACTORIELLE

- Conditions d'ensemble vérifiées ?
  - ✓ Ensemble fini : 10 valeurs uniquement
  - ✓ Valeurs uniques : les valeurs de la factorielle pour n de 1 à 10 sont bien toutes différentes

- On peut utiliser ce concept mathématique pour formaliser le problème
- Définition d'un tableau contenant ces valeurs

TABLEAU DES 10 PREMIÈRES VALEURS DE LA FACTORIELLE

- Déclaration :
  - Fact10 : tableau [10] de entier

- Les valeurs contenues dans le tableau sont indéterminées
  - **Procédure d'initialisation**

- Attention : par définition, les tableaux seront passés en ***Données/Résultats***
  - c'est à dire que les modifications des entrées du tableaux seront conservées après l'exécution de la fonction ou de la procédure

RELATION D'APPARTENANCE

- Test booléen : renvoie vrai ou faux
- Répond à la question : la valeur x appartient-elle à Fact10 ?

- Pour répondre à cette question, la valeur x sera comparée aux éléments contenus dans le tableau Fact10 jusqu'à :
  - soit trouver un élément dont la valeur est égale à x,
    la valeur x appartient à Fact10
  - soit tous les éléments ont été comparés à x
    et aucun n'est égal,
    la valeur x n'appartient pas à Fact10

RELATION D'APPARTENANCE

- Amélioration : on s'arrête dès qu'on trouve une valeur supérieure à celle recherchée
  - Car les valeurs de la factorielle sont rangées dans l'ordre croissant dans le tableau :
    factorielle(n)<factorielle(n+1) pour tout n
  - Le tableau est donc trié
- Définir la relation d'appartenance revient donc à chercher l'élément dans le tableau

LA RELATION D'APPARTENANCE : ALGORITHME

```
fonction appartientAFact10( Fact10 : tableau[10] de entier, x : entier) : booléen
    Données : x
    Données / Résultat : Fact10
    Préconditions : aucune
    description : teste si l'entier x appartient au tableau
    Variable locale : i : entier
        début
        i ← 0
        Tant Que i < 10 Faire
            Si Fact10[i] = x Alors
                Retourner Vrai
            Fin Si
            i ← i + 1
        Fin Tant Que
        Retourner Faux
    Fin
```

LA RELATION D'APPARTENANCE : ALGORITHME

- Ici, on ne réécrit pas l'algorithme avec une boucle POUR à la place du TANT QUE :
  - Le nombre maximum d'itérations est connu (10)
  - Mais il est possible de sortir avant la fin si on trouve l'élément

- Variante : on sort de la boucle dès qu'on dépasse la valeur recherchée
  - Condition supplémentaire dans le "tant que"

LA RELATION D'APPARTENANCE : ALGORITHME

fonction appartientAFact10(Fact10 : tableau[10] de entier, x : entier) : booléen
Données : x
Données / résultat : Fact10
Préconditions : aucune
Description : teste si l'entier x appartient au tableau
Variable locale : i : entier
début
i ← 0
Tant Que (i < 10) et (Fact10[i]<= x) **Faire**
Si Fact10[i] = x **Alors**
Retourner Vrai
Fin Si
i ← i + 1
Fin Tant Que
Retourner Faux
Fin

EXTENSION DU PROBLÈME

- Si on voulait maintenant les 15 premières valeurs de la factorielle
- Faut-il réécrire la fonction d'appartenance ?
  - Seule la taille du tableau change,
    - dans la déclaration
    - dans le test d'arrêt de la boucle
- Paramétrer !
  - La taille du tableau si balayage complet
  - Indices de début et de fin, pour travailler sur une partie du tableau

APPARTENANCE PARAMÉTRÉE

fonction appartientA (T : tableau[100] de entier, n : entier, x : entier) : booléen

Données : x, n (n: nombre de cases occupées dans le tableau)

Données /résultat : T

Préconditions : 100 > n > 0

Description : teste si l'entier x appartient au tableau

Variable locale : i : entier

Début

i ← 0

Tant Que (i < n) et (T[i]<= x) **Faire**

Si T [i] = x **Alors**

Retourner Vrai

Fin Si

i ← i + 1

Fin Tant Que

Retourner Faux

Fin

LES TABLEAUX EN C

- Déclaration :

  type T[dimension]; //tableau à 1 dimension
  type T[ligne][colonne]; //tableau à 2 dimensions

- Opérations sur le tableau :
  - Aucune à part initialisation (limitation du C/C++)
- Opérations sur un élément :
  - Un élément T[i] est une variable,
    les mêmes opérations sont disponibles.
- Utilisation comme paramètre :
  - Identique à la déclaration

LIMITATIONS DU C/C++

- C/C++ ne permet ni de renvoyer plusieurs valeurs,
  ni de renvoyer un tableau
  ➔ uniquement des types de retour simples (entier, réel, booléen)

- Transformer les fonctions concernées (plusieurs résultats ou tableau) en procédures et utiliser des paramètres résultats supplémentaires. (cf. CM4)

CONCLUSION

- Structure de données tableau
  - 2 dimensions
  - N dimensions
  - De n'importe quoi
- Notion d'ensemble mathématique modélisé dans un tableau
- Algorithmes de bases