**Langage de programmation avancé**

**HAUTE ÉCOLE DE NAMUR-LIÈGE-LUXEMBOURG**

**DA-IA – bloc 1**

Atelier 0 – Tableaux à deux indices

Objectifs

* Déclarer et utiliser des tableaux à deux indices
* Utiliser des tableaux comme paramètres de fonction

[Introduction 1](#_Toc81916631)

[A. Tableaux à deux dimensions 2](#_Toc81916632)

[B. Tableau de chaînes de caractères 6](#_Toc81916633)

# Introduction

Dans cet atelier, vous allez apprendre à déclarer et manipuler des tableaux à deux indices.

Dans ce document, plusieurs conventions sont utilisées :

* les mots gras désignent des termes de vocabulaire liés à l’**informatique en général**.
* les mots soulignés et gras désignent des termes de vocabulaire directement liés aux cours de **programmation**.
* le logo signifie que vous avez quelque chose à réaliser.
* le logo est associé aux cadres présentant certaines conventions.
* le logo est associé aux cadres présentant les éléments liés à la propreté/lisibilité du code (*clean code*).

# Tableaux à deux dimensions

Les cellules d'un tableau à une dimension sont indicées de manière consécutive. La représentation graphique du tableau se fait à l'aide d'une suite de cellules les unes contre les autres. Cependant, il est parfois nécessaire de disposer les cellules d'un tableau selon des grilles (tableaux à deux dimensions), des cubes (tableaux à trois dimensions), etc.

En effet, l'algorithmique étant à notre service pour gérer des problèmes divers, il est nécessaire de prévoir une modélisation plus complexe que la simple juxtaposition de cellules.

Si vous devez modéliser un jeu de dames ainsi que le déplacement des pions sur le damier, avec les outils que vous avez abordés jusqu'ici, la solution est de modéliser le damier sous la forme d’un tableau. Chaque cellule du tableau correspond ainsi à une case du damier : les cellules d'indice 0 à 9 pour la première ligne, 10 à 19 pour la deuxième ligne, et ainsi de suite jusqu’à représenter les 100 cases du damier.

Sur base de cette représentation, vous pouvez écrire un algorithme permettant de simuler le jeu de dames. Mais le fait d'utiliser un tableau à simple indice ne facilite pas la clarté de l’algorithme. Il est évidemment plus simple de modéliser un jeu de dames à l'aide d'un damier. L'algorithmique offre la possibilité d'utiliser des tableaux dans lesquels les valeurs ne sont pas repérées par un seul, mais par plusieurs **indices**. Il s'agit alors d'un tableau à plusieurs dimensions.

Plus de 3 dimensions…

Il est intéressant de noter qu'il n'y a pas de limite au nombre de dimensions. Si vous avez compris le principe des tableaux à deux dimensions, il n’y a aucun problème à passer au maniement de tableaux à trois dimensions, quatre dimensions ou plus. Ainsi, l'accès à un élément dans un tableau à trois dimensions se fait à l'aide de trois indices.

Le principal obstacle au maniement systématique de ces tableaux à plus de trois dimensions est que le programmeur, quand il conçoit son algorithme, a pour habitude de dessiner les structures de données qu'il utilise afin de se les représenter.

Or, il est facile d’imaginer concrètement un tableau à une dimension, cela reste faisable pour deux dimensions, cela devient plus complexe pour les tableaux à trois dimensions, mais les tableaux à plus de trois dimensions sont impossibles à visualiser. En effet, l’esprit humain a des difficultés à se représenter les choses dans l’espace, et est souvent perdu dans l’hyperespace (espace comptant plus de trois dimensions). Donc, pour des raisons uniquement pratiques, les tableaux à plus de trois dimensions sont rarement utilisés.

Dans la suite de cet atelier, seuls les tableaux à deux dimensions sont abordés.

## Déclaration

Pour déclarer un tableau à deux dimensions, il faut donc respecter la syntaxe suivante :

<element\_type> <array\_name>[<nbr\_lines>][<nbr\_columns>];

Il faut y remplacer

* <element\_type> par le type qu’auront les éléments du tableau,
* <array\_name> par le nom du tableau,
* <nbr\_lines> par le nombre de lignes du tableau, et
* <nbr\_columns> par le nombre d'éléments dans chaque ligne du tableau, .

Attention ! En mémoire centrale, les cellules sont en fait allouées de façon contigüe. On réserve donc <nbr\_lines> \* <nbr\_columns> \* sizeof(<element\_type>)octets consécutifs. De ce fait, il est impossible, dans le cas de l’allocation statique, d’avoir des lignes de tailles différentes.

Dans le cadre d'un jeu de rôle, le maître du jeu désire pouvoir comparer les caractéristiques des personnages de chacun des joueurs avec lesquels il joue. Chaque personnage a 6 caractéristiques, présentées dans le même ordre : force, dextérité, constitution, intelligence, sagesse et charisme. Dans un nouveau fichier source caractéristiques.c, déclarez le tableau qui permet de mémoriser les caractéristiques de trois personnages.

Pour rester propre, utilisez des constantes symboliques pour les valeurs constantes.

Vous devriez avoir les instructions suivantes.

#define NB\_PERSONNAGES 3

#define NB\_CARACTS 6

int statistiques[NB\_PERSONNAGES][NB\_CARACTS];

Conceptuellement, et donc dans le cadre du cours de « Principes de programmation », ce tableau ressemble à ce qui est présenté ci-dessous.

statistiques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Cependant, en C, puisque les cellules sont consécutives, le tableau se présente comme montré ci-dessous en mémoire centrale. Il s’agit en fait d’un **tableau de tableaux**. À chaque joueur est associé un tableau de caractéristiques.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| statistiques |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Initialisation lors de la déclaration

Comme pour les tableaux simples, il est possible d’initialiser les tableaux à deux dimensions lors de leur déclaration.

La syntaxe est la même mis à part que, comme il s’agit d’un tableau de tableaux, il faut regrouper les valeurs de chaque « ligne » en les entourant par des accolades {} et en séparant ces groupes par des virgules ,.

Dans le fichier source caractéristiques.c, initialisez le tableau des caractéristiques lors de sa déclaration.

Vous devriez avoir une solution semblable à la suivante.

int statistiques[NB\_PERSONNAGES][NB\_CARACTS] = {

{ 8, 14, 12, 10, 13, 18 },

{ 16, 16, 12, 13, 8, 10 },

{ 13, 8, 14, 10, 18, 12 }

};

Il est à noter que lorsque le tableau est initialisé lors de la déclaration, le nombre de lignes est facultatif. Il en est de même dans la signature et les prototypes des fonctions comportant un tableau à deux dimensions.

En mémoire, vous aurez l’équivalent de ce qui présenté ci-dessous.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8  statistiques | 14 | 12 | 10 | 13 | 18 | 16 | 16 | 12 | 13 | 8 | 10 | 13 | 8 | 14 | 10 | 18 | 12 |

## Atteindre un élément précis du tableau

À nouveau, le nombre qui, au sein d’un tableau, sert à repérer chaque valeur s’appelle l’**indice**. Cet indice commence toujours à 0. Un tableau comprenant N éléments voit son indice aller de 0 à N-1.

Comme il s’agit de tableau de tableaux, il faut d’abord choisir la ligne et ensuite la colonne dans cette ligne.

En toute généralité, il faut citer le nom du tableau dont vous désirez atteindre une valeur et le faire suivre de crochets [] avec l’indice du tableau dans lequel vous allez ensuite choisir une cellule via son indice entre crochets [].

La syntaxe est donc la suivante.

<array\_name>[<i\_line>][<i\_column>]

Il faut y remplacer

* <array\_name> par le nom du tableau,
* <i\_line> par l’indice de la cellule correspondant à la ligne,
* < i\_column > par l’indice de la cellule correspondant à la colonne.

Dans l’exemple ci-dessus, la troisième caractéristique du deuxième personnage est donc dans la cellule statistiques[1][2].

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8  statistiques | 14 | 12 | 10 | 13 | 18 | 16 | 16 | 12 | 13 | 8 | 10 | 13 | 8 | 14 | 10 | 18 | 12 |

statistiques[1][2]

Dans le fichier source caractéristiques.c, modifiez la sagesse du deuxième personnage en la faisant passer à 10.

Vous devriez avoir une solution semblable à la suivante.

statistiques[1][4] = 10;

## Affichage et lecture

Les principes sont les mêmes que pour un tableau à une dimension.

Dans le fichier source caractéristiques.c, écrivez les instructions pour choisir une caractéristique d’un personnage et affichez la valeur de cette caractéristique.

Commencez par réfléchir aux fonctions dont vous avez besoin : une fonction pour obtenir un numéro de personnage valide, une fonction pour obtenir un numéro de caractéristique valide…

Concernant la syntaxe, en supposant que l’indice du personnage choisi est dans iPersonnage et que l’indice de la caractéristique est dans iCaract, les instructions permettant d’obtenir et d’afficher la valeur en question sont les suivantes.

scanf("%d", **&**statistiques[iPersonnage][iCaract]);

printf("%d", statistiques[iPersonnage][iCaract]);

Ajoutez une fonction permettant d’imprimer les caractéristiques de chaque joueur.

La solution doit se rapprocher de ce qui est proposé ci-dessous.

void afficheCaracteristiques(int stats[][NB\_CARACTS], int nbPersonnages) {

for(int iPersonnage = 0; iPersonnage < nbPersonnages; iPersonnage ++) {

printf("Personnage %d : ", iPersonnage+1);

for(int iCaract = 0; iCaract < NB\_CARACTS; iCaract++) {

printf("\t%d", stats[iPersonnage][iCaract]);

}

printf("\n");

}

}

# Tableau de chaînes de caractères

Une chaîne de caractères étant un tableau, un tableau de chaînes de caractères est un tableau de caractères à deux dimensions.

Les principes sont les mêmes que pour les chaînes de caractères et les tableaux à deux dimensions…

Dans le fichier source caractéristiques.c, déclarez un tableau permettant de mémoriser les libellés des caractéristiques d'un joueur : force, dextérité, constitution, intelligence, sagesse et charisme.

Avant tout, il faut déterminer le nombre de lignes et le nombre de colonnes.

Vous devriez arriver à l’instruction suivante.

char caracteristiques[6][13];

Faites de même mais, cette fois en initialisant le tableau avec les chaînes de caractères en question. La solution est proposée ci-dessous.

char caracteristiques[][13] = {

"force", "dextérité ",   
 "constitution", "intelligence",   
 "sagesse", "charisme"

};