Université de Relizane Année Universitaire 2021/2022

Département d’Informatique 2 ère année

Module : Algorithmiques et structures de données 3

**FICHE TP : Les tableaux**

**La notion de tableau**

On appelle *tableau* une variable composée de données de même type, stockée de manière contiguë en mémoire (les unes à la suite des autres).

Un tableau est donc une suite de cases (espace mémoire) de même taille. La taille de chacune des cases est conditionnée par le type de donnée que le tableau contient.
Les éléments du tableau peuvent être :

* des données de type simple : int, char, float, long, double...
(la taille d'une case du tableau est alors le nombre d'octets sur lequel la donnée est codée)
* des pointeurs (objets contenant une adresse mémoire. Ce type d'entité sera expliqué dans les chapitres suivants)
* des tableaux
* des structures

Voici donc une manière de représenter un tableau :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| donnée | donnée | donnée | ... | donnée | donnée | donnée |

* Lorsque le tableau est composé de données de type simple, on parle de *tableau monodimensionnel* (ou *vecteur*)
* Lorsque celui-ci contient lui-même d'autres tableaux on parle alors de *tableaux multidimensionnels* (aussi *matrice* ou *table*)

**Les tableaux unidimensionnels**

**Déclaration**

Un tableau unidimensionnel est un tableau qui contient des éléments simples (des éléments qui ne sont pas des tableaux). Un tableau unidimensionnel est donc une suite de « cases » de même taille contenant des éléments d'un type donné (de la longueur de la case en quelque sorte).
Un tableau contenant des entiers peut se représenter de la façon suivante :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| int | int | int | ... | int | int | int |

En langage C, la syntaxe de la définition d'un tableau unidimensionnel est la suivante :

type Nom\_du\_tableau [Nombre d'éléments]

* *type* définit le type d'élément que contient le tableau (rappel : un tableau en langage C est composé uniquement d'éléments de même type), c'est-à-dire qu'il définit la taille d'une case du tableau en mémoire
* *Nom\_du\_tableau* est le nom que l'on décide de donner au tableau, le nom du tableau suit les mêmes règles qu'un nom de variable
* *Nombre d'éléments* est un nombre entier qui détermine le nombre de cases que le tableau doit comporter

Voici par exemple la définition d'un tableau qui doit contenir 8 éléments de type char :

char Tableau [8]

**EXERCICE 1**

Supposons que nous avons le tableau suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| 10 | 23 | 505 | 8 |

 Ecrire un programme qui permet :

* De déclarer ce tableau
* De remplir le tableau
* D’afficher les éléments de ce tableau.
* D’initialiser toutes les valeurs de ce tableau à 0. Afficher les nouvelles valeurs pour vérifier.

**EXERCICE 2**

Ecrire un algorithme qui détermine la plus grande et la plus petite valeur dans un tableau d’entiers A. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

Solution

Exercice 01

int main(int argc, char \*argv[])

{

 int tableau[4], i = 0;

 tableau[0] = 10;

 tableau[1] = 23;

 tableau[2] = 505;

 tableau[3] = 8;

 for (i = 0 ; i < 4 ; i++)

 {

 printf("%d\n", tableau[i]);

 }

 return 0;

}

Initialisation et vérification

int main(int argc, char \*argv[])

{

 int tableau[4], i = 0;

 // Initialisation du tableau

 for (i = 0 ; i < 4 ; i++)

 {

 tableau[i] = 0;

 }

 // Affichage de ses valeurs pour vérifier

 for (i = 0 ; i < 4 ; i++)

 {

 printf("%d\n", tableau[i]);

 }

 return 0;

}

Exercice 02

Solution

#include<stdio.h>

#define

void main()

{

 int n=5,i=0,t[n];

 t[0]=3;

 t[1]=1;

 t[2]=1;

 t[3]=5;

 t[4]=5;

 int imax=-1,imin=-1,max=t[0];

 int min=t[0];

 while(i<n)

 {

 printf(“%d | “,t[i]);

 i++;

 }

printf(“\n”);

i=0;

 while(i<n)

 {

 if (t[i]>max){ max=t[i]; imax=i;}

 if(t[i]<min) {min=t[i]; imin=i;}

 i++;

 }

printf(“le max est %d d’indice %d \n”,max,imax);

printf(“le min est %d d’indice %d \n”,min,imin);

}