UNIVERSITÉ ABOU BAKR BEL-KAID–TLEMCEN
FACULÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE
Première année licence



Algorithmique et structure de données 1

Chapitre 5:Les tableaux et les chaines de caractères

Mme HABRI née BENMAHDI Meryem Bochra

Année universitaire: 2024-2025

Plan du cours

1. **Introduction : nécessité de données tabulaires (exemples)**
2. **Tableau**
3. **Tableau en mémoire**
4. **Tableau à une dimension (1D)**
5. **Le tableau à deux dimensions (2D), ou matrice**
6. **Quelques applications des tableaux 1D et 2D**
7. **Le type chaîne de caractères (cas particulier de tableau 1D)**
8. **Quelques applications des chaînes de caractères**

Introduction

Exercice 1: Écrire un programme permettant de saisir les 7 moyennes des matières du premier semestre d'un étudiant inscrit en première année informatique puis afficher avec sa moyenne.

Exercice 1

```
#include <stdio.h>
int main() {
    float note, somme = 0, moyenne;
    int i;
    // Saisie et accumulation des 7 moyennes
    for (i = 1; i <= 7; i++) {
        printf("Entrez la moyenne de la matiere %d : ", i);
        scanf("%f", &note);
        somme += note;
    }
    // Calcul de la moyenne générale
    moyenne = somme / 7;
    // Affichage du résultat
    printf("\nLa moyenne generale de l'etudiant est : %.2f\n",
moyenne);
    return 0;
}
```

Introduction

Exercice 2: Écrire un programme permettant de calculer la moyenne générale de 600 étudiants. Pour cela, le programme doit :

1. **Saisir la moyenne de chaque étudiant (une moyenne comprise entre 0 et 20).**
2. **Vérifier que chaque moyenne saisie est valide. Si la valeur saisie est inférieure à 0 ou supérieure à 20, afficher un message d'erreur et demander à l'utilisateur de saisir à nouveau la moyenne.**
3. **Calculer la somme des moyennes des 600 étudiants.**
4. **Calculer et afficher la moyenne générale des 600 étudiants avec deux chiffres après la virgule.**

Travail demandé : Écrire un programme en langage C utilisant les structures répétitives for et while pour résoudre ce problème.

Exercice 2

```
#include <stdio.h>
int main() {
   float note, somme = 0, moyenne;
   int i;
   printf("=== Calcul de la moyenne generale des 600 etudiants ===\n");
   // Boucle de saisie pour les 600 étudiants
   for (i = 1; i <= 600; i++) {
      printf("Entrez la moyenne de l'etudiant %d : ", i);
      scanf("%f", &note);
      // Contrôle de saisie (note entre 0 et 20)
      while (note < 0 || note > 20) {
         printf("Erreur ! La note doit etre entre 0 et 20.\n");
         printf("Reentrez la moyenne de l'etudiant %d : ", i);
         scanf("%f", &note); }
      somme += note;  // ajout de la note à la somme totale}
// Calcul de la moyenne générale
   moyenne = somme / 600;
   // Affichage du résultat
   printf("\nLa moyenne generale des 600 etudiants est : %.2f\n", moyenne);
   return 0;}
```

Introduction

Exercice 3: Écrire un programme qui permet de saisir les moyennes de tous les 600 étudiants inscrit en première année licence informatique puis de déterminer combien d'entre elles sont supérieures à la moyenne de la promotion.

Solution

```
void main() { int M1,
M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13,..... ...
MoyenneGenerale =
(M1+M2+M3+M4+M5+M6+M7+M8+M9+M10+M11+M12+
M13+...)/600;
 printf("La moyenne est : %d", Moyenne); Euuuuuuuuuuuuuhhhhh
!!!! ... }
```

N'y a t-il pas un moyen plus simple et plus élégant pour solutionner ça?

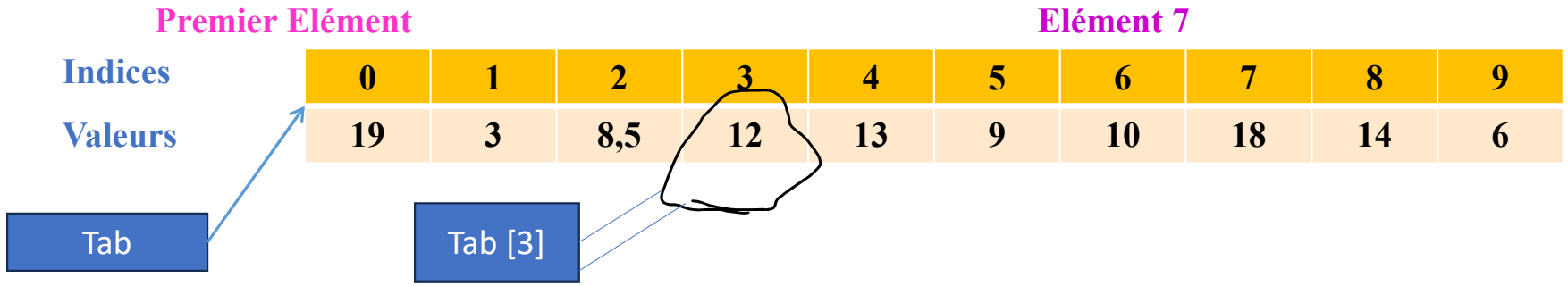
Nécessité des tableaux

- **Les tableaux, c'est la solution idéale.**
- **Lorsqu'un programme doit traiter plusieurs valeurs similaires (notes, températures, etc.)**
- **Il devient difficile d'utiliser uniquement des variables simples.**
- **Les tableaux permettent de regrouper des données du même type sous un seul nom.**

Exemple : stocker les notes de 600 étudiants dans un tableau.

Les tableaux

- **Un ensemble de valeurs portant le même nom de variable et repérées par un nombre, s'appelle un tableau, vecteur, ou encore une variable indicée.**
- **Le nombre qui, au sein d'un tableau, sert à repérer chaque valeur s'appelle l'indice, L'indice représente la position d'un élément dans le tableau**
- **Chaque élément du tableau est désigné par le nom du tableau, suivi de l'indice de l'élément, entre crochets.**



Le tableau en mémoire

- **Un schéma d'illustration d'un tableau de 10 cases en mémoire qui commence à l'adresse 2001.**
- **Lorsqu'un tableau est créé, il prend un espace contigu en mémoire : les cases sont liées les unes à la suite des autres.**
- **Toutes les cases d'un tableau sont du même type. Ainsi, un tableau de int contiendra uniquement des int, et pas autre chose.**

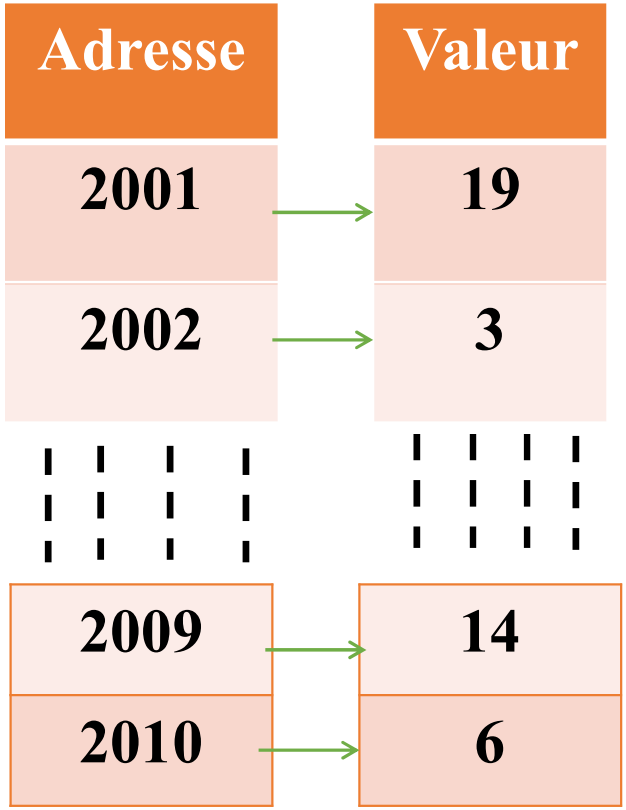


Tableau á une seule dimension (1D)
Déclaration

Définition: **Un tableau 1D est une collection d'éléments de même type.**

Déclaration :

Exemple:

En algorithmique:

```
Var  taille: entier
     varNom: tableau [1…taille] de type
```

En langage C

```
int taille;
Type varNom[taille];
```

En algorithmique:

```
note : tableau [1..10] de réel
note[1]←12,5
```

En langage C

```
float note[10 ];
note[0]=12,5;
```

Il suffit donc de rajouter entre crochets le nombre de cases que vous voulez mettre dans votre tableau. pas de limite.

Tableau á une seule dimension (1D)

Accès aux éléments du tableau

Pour accéder à chaque élément du tableau, il faut écrire le nom du tableau suivi de l'indice de l'élément concerné entre crochet.

En algorithmique:

note[1]←12,5

En langage C

Note[0]=12;

Remarques

- **En programmation C, un tableau commence à l'indice numéro 0! Notre tableau Note de 10 réels a donc les indices 0, 1, 2,... et 9. Il n'y a pas d'indice 10 dans un tableau de 10 cases! C'est une source d'erreurs très courantes pour les débutants.**
- **En algorithmique, le tableau commence à l'indice numéro 1 ! Notre tableau Note de 10 réels a donc les indices 1, 2,... et 10.**

Tableau á une seule dimension (1D)
initialisation d'un tableau

Pour initialiser un tableau dans un algorithme

```
Var  note : tableau [1..5] de réel
        Début
note[1]←12,5
note[2]←10,5
note[3]←19
note[4]←10
note[5]←15,5
        Fin
```

Tableau á une seule dimension (1D)
initialisation d'un tableau

Pour initialiser un tableau en programmation C : il existe deux méthodes pour initialiser un tableau

```
void main()
 { int T[4];
T[0]=2;
T[1]=8;
T[2]=3;
T[3]=1;}
```

```
void main()
 { int T[4] = {2, 8, 3, 1};
 double note[3] = {10, 16, 14.892};
 int tab[6] = {10, 23}; // 10, 23, 0, 0, 0, 0
}
```

- **On peut initialiser tous les éléments d'un tableau par un zéro.**
- **On peut également initialiser que les premiers éléments d'un tableau, tous ceux qui ne sont pas renseignées, seront automatiquement mis à 0.**

Tableau á une seule dimension (1D)

parcourir un tableau

- **Supposons qu'on veuille maintenant afficher les valeurs de chaque case du tableau.**
- **On aurait pu faire autant de printf qu'il y a de cases. Mais ce serait répétitif et lourd, et imaginez un peu la taille de notre code si on devait afficher le contenu de chaque case du tableau une à une!**
- **Le mieux est de se servir d'une boucle qui est très pratique pour parcourir un tableau :**

Exemple

```
#include<stdio.h>
int main()
{double notes[6]={15.5, 13, 4, 19, 17.5, 16};
for (int i = 0 ; i <6; i++)
printf("%f\n", notes[i]);
return 0;}
```

Tableau á une seule dimension (1D)

parcourir un tableau

```
#include<stdio.h>
 void main()
 {double notes[6]={15.5, 13, 4, 19, 17.5, 16};
 for (int i = 0 ; i < 8 ; i++)
 printf("%f\n", notes[i]);}
```

- **La boucle parcourt le tableau à l'aide d'une variable appelée i (c'est le nom le plus souvent utilisé pour parcourir un tableau!).**
- **Notez qu'on peut mettre une variable entre crochets pour parcourir le tableau, c'est-à-dire accéder à ses valeurs.**
- **Attention à ne pas tenter d'afficher la valeur de notes[7]!Sinon vous aurez soit n'importe quoi, soit une belle erreur, l'OS interrompra votre programme car il aura tenté d'accéder à une adresse ne lui appartenant pas.**

si on revenait à l'exercice 3?

Exercice 3: Trouver l'organigramme et écrire l'algorithme et le programme qui permet de saisir les moyennes de tous les 600 étudiants inscrit en première année licence informatique puis de déterminer combien d'entre elles sont supérieures ou égale à la moyenne de la promotion.

Exercice d'application

Écrire un programme qui lit un tableau (1D)de nombres réels, puis détermine et affiche le nombre de fois qu'une valeur donnée apparaît dans ce tableau.