

Tableaux en C

Principaux algorithmes
de programmation
des tableaux en C



Plan

- ♦ Rappels de la définition
- ♦ Programmer des tableaux en C
 - ♦ Affichage
 - ♦ Lecture
 - ♦ Tri



Tableaux késaco ?

- ◆ Définition informatique
 - ◆ Collection d'objets de même type
 - ◆ Groupement sous un même identificateur
 - ◆ Accès individuel par indice entier
 - ◆ Un élément d'un tableau peut aussi être un tableau
- ◆ Exemples d'usage
 - ◆ Liste des étudiants
 - ◆ Liste des départements français
 - ◆ Jeu de cartes
 - ◆ Tirage du Loto
 - ◆ Plateau de jeu : échiquier/damier
 - ◆ Vecteur en géométrie
 - ◆ Matrice en algèbre



Syntaxe des tableaux C

◆ Déclaration

- ◆ `type identificateur[taille];`

◆ Accès à un élément

- ◆ `X = identificateur[indice];`

◆ Modification d'un élément

- ◆ `identificateur[indice] = X;`

◆ Plage des indices : [0..taille-1]

◆ Initialisation

- ◆ `type id[] = {val0,...,valn-1};`

◆ Exemples

- ◆ `int loto[6];`

- ◆ `loto[0] = 27;`

- ◆ `char mot[10];`

- ◆ `'r' == mot[9];`

- ◆ `float vecteur[3] = {1.0,0.0,-2.5};`

- ◆ `vecteur[0] = 4.5;`

- ◆ `int echecs[8][8];`

- ◆ `echecs[2][5] = 3;`



Affichage d'un tableau

```
void table_ecrire(int table[], int n)
{
    /* variable de parcours des éléments de la table */
    int i;

    /* Affichage des n-1 premiers éléments,
       séparés par des virgules */
    for (i=0; i<n-1; i++)
        printf("%i, ", table[i]);

    /* Affichage du dernier élément de la table,
       suivi d'un passage à la ligne */
    printf("%i\n", table[i]);
}
```

Lecture d'un tableau

```
void table_lire(int table[], int n)
{
    /* variable de parcours des éléments de la table */
    int i;

    /* Affichage des n éléments,
       séparés par des espaces */
    for (i=0; i<n; i++)
        scanf("%i",&table[i]);
}
```



Tri d'un tableau

- ♦ Trier un tableau, c'est ordonner ces éléments
 - ♦ Nécessite une relation d'ordre sur les éléments
 - ♦ Ordre croissant ou décroissant
- ♦ Exemples
 - ♦ Non-trié : {4,3,7};
 - ♦ Trié : {3,4,7} dans l'ordre croissant des entiers
 - ♦ Non-trié : {'c','e','b','d'}
 - ♦ Trié : {'b','c','d','e'} dans l'ordre alphabétique



Algorithmme de tri

- ◆ Nombreuses méthodes de tri appelées algorithmes de tri
- ◆ Une méthode simple le tri-bulle :
 - ◆ Échange de deux éléments successifs si ils ne sont pas ordonnés
 - ◆ Répétition tant qu'il y a des échanges
- ◆ Exemples de tri-bulle :
 - ◆ 1er tour
 - ◆ $\{3,2,9,5\} \Rightarrow$ Permutation
 $\{2,3,9,5\}$
 $\{2,3,9,5\} \Rightarrow$ Permutation
 - ◆ 2ème tour
 - ◆ $\{2,3,5,9\}$
 $\{2,3,5,9\}$
 $\{2,3,5,9\}$
 - ◆ Résultat : $\{2,3,5,9\}$



Programme de tri bulle

```
void table_trier(int table[], int n){
    int i, tmp, permutation = 1;

    while (permutation == 1){
        permutation = 0;
        for (i=0; i<n-1; i++){
            if (table[i] > table[i+1]){
                tmp = table[i];
                table[i] = table[i+1];
                table[i+1] = tmp;
                /* Il y a eu permutation */
                permutation = 1;
            } /* if */
        } /* while */
    }
}
```



Conclusions

- ◆ Tableaux servent à représenter une collection de données
- ◆ Accès individuel par indice entier
- ◆ Programmation en C :
 - ◆ Déclaration : `type table[n];`
 - ◆ Indices de 0 à $n-1$
 - ◆ Utilisation : `x = table[ind];` ou `table[ind] = x;`

