

Structures de données complexes

✓ Structure (enregistrement)

- » Une structure permet de regrouper des variables de type différent
- » Définition d'un type

```
struct date
{
    int jour;
    int mois;
    int an;
};
struct date d1, d2;
```

```
struct personne
{
    char nom[20];
    char prenom[20];
    int age;
};
struct personne p;
```

» Initialisation

```
struct date d = {7, 9, 2005};
```

» Accès à un champ

```
d.jour++;
if (d.mois > 12) {d.mois = 1; d.an++;}
```

Structures de données complexes

» Composition de structures

- o Une structure est dite imbriquée lorsque l'un de ses membres est une structure

```
struct date
{
    int jour;
    int mois;
    int an;
};
struct date d1, d2;
```

```
struct personne
{
    char nom[20];
    char prenom[20];
    struct date naissance;
};
struct personne p;
```

o Initialisation

```
struct personne jean = { "Dupont", "Jean", {3, 10, 1985} };
```

o Accès à un champ

```
jean.naissance.an = 1985;
```

Structures de données complexes

» Tableau de structures

- o Déclaration
 - Faire précéder le nom du tableau par le nom de la structure

```
struct date
{
    int jour;
    int mois;
    int an;
};
```

```
struct date tab[10];
tab[0].jour = 12;
tab[0].mois = 2;
tab[0].an = 2011;
```

» Structures récursives

- o Structure qui fait appel à elle-même dans sa définition

```
struct cellule
{
    int val;
    struct cellule *suiv;
};
```

Exercice

✓ Exercice

- » Un relevé de températures journalier contient la température minimale et la température maximale de la journée. Définir une structure "relevé" permettant de caractériser un relevé de températures.
- » Écrire un programme C permettant de saisir puis d'afficher les caractéristiques d'un relevé de températures
- » Définir un tableau de relevés de 10 cases
- » Écrire un programme C permettant de saisir puis d'afficher le tableau de relevés

Exercices

✓ Tri bulle

- » On souhaite écrire un programme C qui
 - o permet de trier les nb valeurs contenues dans le tableau (supposé initialisé) selon l'ordre croissant en utilisant la méthode du "tri bulle"
 - o Principe de la méthode du "tri bulle"
 - Il consiste à parcourir la suite des éléments à trier de l'indice 0 à l'indice nb-1 (nb désignant le nombre d'éléments à trier) en intervertissant toute paire d'éléments consécutifs (T[j-1], T[j]) non ordonnés. Après un parcours, l'élément maximum se trouve en position T[nb-1]
 - Le parcours du tableau est effectué nb-1 fois (1^{er} parcours en considérant les éléments de 0 à nb-1, le 2nd en considérant les éléments de 0 à nb-2, ..., le n-1^{ème} parcours en considérant les éléments de 0 à 1).
 - Le nom "tri bulle" vient du fait que les grands nombres se déplacent vers la droite en poussant les bulles successives de la gauche vers la droite.

Exercices

- o Définir une structure tableau qui aura pour premier champ le tableau d'entiers et comme second champ le nombre de cases effectivement remplies parmi les MAX de ce tableau
- o Écrire un programme C qui
 - demande à l'utilisateur le nombre nbE (0 < nbE ≤ MAX) de valeurs sur lesquelles il souhaite travailler
 - permet à l'utilisateur de saisir au clavier nbE valeurs et de les ranger dans un tableau d'entiers de type struct tableau
 - affiche à l'écran le contenu du tableau d'entiers rempli. Il ne s'agira pas d'afficher la totalité du tableau mais uniquement les nbE cases remplies
 - permet de trier les nbE valeurs contenues dans le tableau de type struct tableau selon l'ordre croissant en utilisant la méthode du "tri Bulle"
 - affiche à l'écran le contenu du tableau d'entiers trié

Exercices



Exemple : Si on désire trier un tableau de 4 entiers dans l'ordre croissant

5	2	3	1	$T[0] > T[1]$	on échange $T[0]$ et $T[1]$,
2	5	3	1	$T[1] > T[2]$	on échange $T[1]$ et $T[2]$,
2	3	5	1	$T[2] > T[3]$	on échange $T[2]$ et $T[3]$,
2	3	1	5	L'élément en position 3 est en bonne position.	

Plusieurs passages sur l'ensemble des données sont nécessaires pour l'obtention d'un tri complet

2	3	1	5	$T[0] < T[1]$	on ne fait rien,
2	3	1	5	$T[1] > T[2]$	on échange $T[1]$ et $T[2]$,
2	1	3	5	L'élément en position 2 est en bonne position	

2	1	3	5	$T[0] > T[1]$	on échange $T[0]$ et $T[1]$,
1	2	3	5	L'élément en position 1 est en bonne position, donc celui en position 0 l'est forcément	