

Travaux pratiques n°12 (2h): Structures: Déclaration et utilisation

Objectifs

- Déclarer et manipuler des structures en C.

Documents nécessaires

- sujet du TD12 et notes de TD12
- support de cours sur les structures

Partie 1: Compléments pratiques au langage C

L'instruction *typedef*:

Comme indiqué en cours l'instruction *typedef* permet de définir un nouveau type. Par exemple on peut définir le type entier à partir de *int* ou le type *string* à partir de *char*:

```
typedef int entier; // le type entier est équivalent à int
```

```
typedef char str40[41]; // le type str40 est équivalent à un tableau de 40 char
```

En particulier, on peut éviter le mot *struct* dans les types structures en utilisant *typedef*.

Ainsi:

```
typedef struct
```

```
{  
    int x;  
    int y;  
} point;
```

permet de définir le type *point* équivalent à *struct point* et de déclarer des variables de type *point*:

```
point p1,p2; // déclare deux variables de type point.
```

Vous avez aussi la possibilité de la faire en deux temps:

```
struct point
```

```
{  
    int x;  
    int y;  
};
```

```
typedef struct point point;
```

Partie 2: Traduire des corps d'algorithmes détaillés avec des structures en langage C

Exercice 1.

Traduire et tester en langage C les algorithmes de l'exercice 2 de ce TD.

Exercice 2: Une bataille navale

Considérons un bateau dans un jeu de bataille navale, il a un nom ("bateau_1" par exemple), un type (croiseur, destroyer, sous-marin, etc...), une position sur la grille ((1,1) par exemple) de début et une position de fin ((1,5) par exemple), la longueur initiale du bateau (en nombre de cases), un état (touché x fois, coulé, ou en parfait état).

- a) Donner une structure adéquate pour *bateau*.
- b) Écrire une fonction en C qui a la position d'un tir et un bateau en entrée, qui teste le tir et modifie le bateau s'il a été touché et qui renvoie 0 si le tir a échoué, 1 si le bateau a été touché et 2 s'il a été coulé.
- c) Dans une bataille navale à 2 joueurs, chaque joueur doit disposer d'un certain nombre de bateaux.
Écrire une fonction en C, *initBat* qui retourne un bateau défini par le type et la longueur qui sont donnés en entrée et par le nom et les positions début et fin qui sont demandées à l'utilisateur dans le corps de la fonction.