

Cahier de TD n° 5

THÈME 1 : VRAI/FAUX, DÉFINITION DE STRUCTURES

VRAI/FAUX

MANIPULATIONS DE BASE : LE '!' ET LA '→'

THÈME 2 : FONCTIONS ET STRUCTURES

NOMBRES COMPLEXES $Z = \text{REEL} + I \text{ IMAGINAIRE}$

Thème 1 : Vrai/faux, définition de structures

VRAI/FAUX

- Une structure est un type
- Une structure est un type défini par l'utilisateur
- Une structure a au moins un champ de type entier
- Un champ est une structure
- Un champ comporte des structures
- Une structure comporte un ou des champs
- Un champ est défini uniquement par son type
- Un champ est défini uniquement par son nom
- Un champ peut être de n'importe quel type
- On ne peut pas définir de variable dont le type est une structure
- L'accès à un champ d'une structure se fait par la syntaxe : '*'
- On peut définir un pointeur sur une structure

choisissez, pour les exemples suivants :

les types et les noms des champs pour les objets listés, et écrivez la définition de la structure associée (pas plus de 5/6 champs par structure) :

- une adresse postale
- un CD musical
- un client d'une banque
- un chien
- une émission télévisée
- un joueur de football
- une équipe de football

Manipulations de base : le '.' et la '→'

Dans les exemples de programme suivants, indiquez quelles écritures sont correctes syntaxiquement.

```
typedef struct ordi
{
    long capa; // capacité de la mémoire
    long freq; // fréquence du proco
    char marque[30]; // marque du PC et du processeur
    long hd; // capacité du disque dur
} t_ordi;

int main()
{
    t_ordi mon_pc;
    long hd;

    printf("%o", mon_pc);
    t_ordi.capa = 512;
    mon_pc.marque = "DELL AMD Athlon 2200+";
    mon_pc.freq = 1800;
    hd = 120;
    mon_pc.hd = hd;

    return 0;
}
```

Thème 2 : Fonctions et structures

Les fonctions sont très souvent associées aux structures, pour la simple et bonne raison qu'une structure ne se manipule pas comme une variable de type classique, et il faut donc prévoir des fonctions pour toutes les opérations classiques que l'on doit réaliser.

Nombres complexes $z = \text{reel} + i \text{ imaginaire}$

On veut définir un nouveau type pour représenter et manipuler des nombres complexes.

- Définir le type **Complexe** qui permette de représenter un nombre complexe comme une structure ayant deux champs de type doublenommés **reel** et **imaginaire**.

Les fonctions demandées ci-après doivent avoir en paramètre un pointeur sur une structure de type **Complexe.**

- Définir une fonction d'initialisation d'un complexe **initialiserComplexe** ($z = (0, 0)$).
- Définir une fonction de saisie au clavier d'un complexe **remplirComplexe**.
- Définir une fonction d'affichage d'un complexe **afficherComplexe**.
- Définir une fonction **cmpComplexe** qui compare deux complexes et renvoie 0 s'ils sont égaux et -1 sinon.
- Définir une fonction **sommeComplexe** qui effectue la somme de deux complexes.
- Définir une fonction **produitComplexe** qui effectue le produit de deux complexes.
- Tester les fonctions dans la fonction principale de votre programme.
- Définir des fonctions d'initialisation, de saisie, d'affichage pour un tableau de 10 complexes alloués statiquement.