

Fiche exo 3 : Structures de données

Pour chaque question, on donnera un exemple d'utilisation dans un programme principal.

Exercice 1 : nombres complexes

- a. Définir la structure `Complexe` représentant le nombre complexe $z = a + ib$.

Ecrire les fonctions suivantes :

- b. `PrintC` : affichage d'un nombre complexe ;
- c. `ScanC` : saisie d'un nombre complexe ;
- d. `SumC` : addition de deux complexes ;
- e. `SubC` : soustraction de deux complexes ;
- f. `MulC` : Multiplication de deux complexes.

Exercice 2 : heures

- a. Définir une structure de données `Heure` permettant de représenter une heure au format hh : mm : ss ;

Ecrire les fonctions suivantes :

- b. `PrintH` : affichage d'un élément de type `Heure` passé en paramètre ;
- c. `ScanH` : saisie d'un élément de type `Heure` par l'utilisateur (vérifier la cohérence) ;
- d. `H2S` : conversion d'un élément de type `Heure` en nombre de secondes (entier) ;
- e. `S2H` : conversion d'un nombre de secondes (entier) en un élément de type `Heure` ;
- f. `SumH` : addition de deux durées (éléments de type `Heure`).

Exercice 3 : carnet d'adresse

1. Créer une structure `Contact` contenant deux champs : `nom` et `tel`. Le nom peut contenir 32 caractères et le numéro 10 caractères.

Ecrire les fonctions suivantes :

- a. `ScanContact` : saisie au clavier d'un élément de type `Contact`.
- b. `PrintContact` : affiche les informations contenues dans un élément de type `Contact` passé en argument.
2. Créer une nouvelle structure pour représenter un carnet d'adresses. Cette structure `Carnet` contiendra un tableau de 20 éléments de type `Contact` et un compteur indiquant le nombre de personnes dans le tableau.

Ecrire les fonctions suivantes :

- a. `AddContact` : ajoute un `Contact` dans le `Carnet` (passé en paramètre) et retourne le carnet modifié;
- b. `PrintRep` : affiche le `Carnet` ;
- c. `Alpha` : trie le carnet par ordre alphabétique.