TP n° 1 : langage C (révisions ?)

Exercice 1 (Programme simple ; avec fonction de calcul / de modification)
L'objectif de ce programme :

- saisir deux valeurs (variables a et b), en faisant en sorte que $a < b$
- saisir une troisième valeur (variable x), qui doit être entre a et b
- déterminer de quelle extrémité x est le plus proche, puis donner cette valeur à la variable x.

1°) Réalisez ce traitement sans appel de fonction (tout dans la fonction principale).

2°) Utilisez une fonction pour le calcul de la valeur à attribuer à la variable x (après les saisies).

3°) Utilisez une fonction pour modifier la valeur de x (après les saisies).

Exercice 2 (Structures, passage de paramètre, compilation séparée)
On souhaite pouvoir gérer des variables de type `struct temps`, représentant des durées en heures, minutes et secondes. On envisage différentes fonctionnalités, dont les prototypes sont les suivants :

- `struct temps creer(int h, int m, double s)`
  renvoie une variable de type `struct temps` créée avec les valeurs passées en paramètre
- `_Bool correct(struct temps t)`
  pour s'assurer que les champs ont tous une valeur positive (ou nulle)
- `void afficher (struct temps t)`
  affichage, avec 2 positions pour les minutes et pour la partie réelle des secondes (forme : `2:09:08.3` au lieu de `2:9:8.3`)
- `struct temps normaliser(struct temps)`
  ramène les minutes et les secondes à une valeur strictement plus petite que 60
- `struct temps creer2(void)`
  permet créer un struct temps en affectant aux champs des valeurs lues au clavier
- `void saisir(struct temps *pt)`
  permet à l'utilisateur de renseigner les champs d'un struct temps passé en paramètre

1°) Rassemblez les signatures de ces fonctionnalités au sein du fichier d'en-têtes `temps.h`, avec la définition du type `struct temps`.

2°) Codez ces fonctions (fichier `temps.c`).

3°) Ecrivez un *main* de test (fichier `principal.c`).

4°) Ecrivez un fichier `Makefile`.

5°) Testez l'ensemble de vos développements.

Exercice 3 (Passage de paramètres avec des variables structurées)
On considère des variables A, B, C, ..., contenant un entier et un réel :

par exemple $A \left\{ \begin{array}{c} 7 \\ 3.01 \end{array} \right.$, $B \left\{ \begin{array}{c} 6 \\ 4.3 \end{array} \right.$, $C \left\{ \begin{array}{c} -2 \\ 10.0 \end{array} \right.$, ...

On souhaite ordonner leurs contenus, par ordre croissant de leurs champs entiers.

1°) Dans les cas suivants, peut-on utiliser des variables structurées (sans déclaration d'un type structuré) ?

- tout le code est écrit dans le *main*
- on utilise une/des fonction(s) prenant en paramètre de telles variables

2°) Ecrivez une fonction `echanger` permettant d'échanger les contenus de telles variables.

3°) Ecrivez une fonction `ordonner` permettant d'ordonner les contenus de telles variables selon la valeur de leur champ entier.
Vous pouvez écrire deux versions de cette fonction : sans utiliser la fonction `echange` / en l'utilisant.

4°) Ecrivez une fonction `ordonner3` permettant d'ordonner les contenus trois de ces variables.
NB : Vous pouvez utiliser la fonction `ordonner` (ou pas).