

TP - Structure et tri

Objectif(s)

- ★ Approfondir l'utilisation des structures
- ★ Implanter des algorithmes de tri

Exercices obligatoires

Exercice 1

Au cours de cette exercice, nous allons définir un `struct` contenant des informations sur des personnes, créer un tableau de personnes et ensuite trier ce tableau selon différents critères.

Question 1

Définissez une `struct` `personne` avec un champ `nom`, un champ `prénom`, champ `date de naissance` (qui est lui même un `struct`). Pour chaque chaîne de caractères, on utilisera une taille maximum définie comme une constante `TAILLE_MAX` initialisée à 30.

Question 2

Définissez une fonction vous permettant d'afficher une personne sous la forme :
Nom Prénom (date de naissance). Par exemple, Setrouillard Stéphane (15/04/1975) ou Setrouillard Stéphane (15 avril 1975) si vous avez pu récupérer la `struct` date d'un tp précédent.

Pour tester votre fonction dans un main, afin d'affecter le bon nom dans la structure, il est possible d'utiliser une ligne du type :

```
struct personne p = {"Stephane", "Setrouillard", {15, 4, 1975}};
```

Question 3

Définissez une fonction `lire_personne` vous permettant de créer une personne avec les noms, prénoms et date de naissance entrés au clavier par l'utilisateur.

Question 4

Définissez une fonction `personne_aleatoire` utilisant la fonction de génération de noms disponible sur e-campus pour tirer une personne au hasard. (Vous gerez le tirage de la date de naissance).

Question 5

Définissez une fonction qui prend en argument deux personnes et renvoie 1 si la première personne est classée avant la deuxième par ordre alphabétique en prenant en compte le nom et le prénom, 0 sinon.

Testez votre fonction sur des exemples bien choisis afin de la valider.

Question 6

Définissez une fonction qui prend en argument deux personnes et renvoie 1 si la première personne est plus jeune que la deuxième, 0 sinon.

Testez votre fonction sur des exemples bien choisis afin de la valider.

Question 7

Écrivez une fonction permettant d'avoir un tableau de 20 personnes et une deuxième fonction permettant de l'afficher, avec une personne par ligne.

Question 8

Écrivez une fonction pour trier un tableau de personnes par ordre alphabétique. Vous utiliserez l'algorithme du tri sélection.

Question 9

Écrivez une fonction pour trier un tableau de personnes par age. Vous utiliserez l'algorithme du tri par insertion.

Exercice 2 – Amélioration du tri sélection

Pour positionner la i ème valeur, l'algorithme du tri sélection tel qu'il a été vu s'effectue en deux étapes :

- la recherche de la position $posMini$ de la plus petite valeur à l'étape i
- l'échange des valeurs aux positions i et $posMini$.

L'algorithme du tri sélection peut être amélioré lors de la phase de recherche en recherchant la position de la plus petite et de la plus grande valeur puis en effectuant les échanges pour avoir le début et la fin du tableau triés.

Exemple :

	2	4	1	8	3	5
1	4	2	5	3	8	
1	2	3	4	5	8	
1	2	3	4	5	8	

Certains cas sont à réfléchir sérieusement. Par exemple :

	2	7	1	8	3	4
1	7	2	4	3	8	

Lors de cette étape, 2 est la valeur minimale et 7 la valeur maximale. Il y a potentiellement un problème lors des échanges si on ne fait pas bien les choses.

Implantez cet algorithme sur un tableau d'entiers.

Renforcement**Exercice 3 – le tri rapide**

Le tri rapide est un tri très utilisé en pratique, de par sa bonne complexité en moyenne. La méthode de ce tri est la suivante :

Étape 1 :

1. Le *pivot* est choisi : c'est le premier élément du (sous-)tableau à trier,
2. on fait un balayage vers la gauche du (sous-)tableau à trier en commençant au dernier élément et un balayage vers la droite qui commence à l'élément qui suit le pivot,
3. le balayage gauche s'arrête quand il rencontre un élément supérieur au pivot,
4. le balayage droite s'arrête quand il rencontre un élément inférieur au pivot
5. les deux éléments sont échangés et les balayages continuent de leur point d'arrêt,
6. l'itération finit quand l'indice du balayage gauche dépasse celui du balayage droite,
7. On échange alors le contenu de la case du pivot avec la case contenant le dernier élément inférieur au pivot (juste avant le point d'arrêt). On obtient un tableau de la forme :

L_1	pivot	L_2
-------	-------	-------

 où L_1 contient les éléments inférieurs au pivot (mais pas forcément triés entre eux) et L_2 les éléments supérieurs au pivot.

Étape 2

1. on recommence pour trier L_1 et L_2 , respectivement, en prenant leur premier élément comme pivot.

Une solution simple est d'utiliser un tableau auxiliaire pour l'étape 2 et copier ses contenus dans le tableau d'origine. Une autre solution (plus efficace mais moins simple) consiste à indiquer lors de l'appel du tri les numéros du début et de la fin du tableau, et se limiter au sous-tableau encadré par ces éléments.

Implémentez le tri rapide et testez le.