

esprit Ecole Supérieure Privée d'Ingénierie et de Technologies	Objectif (2) : L'allocation dynamique	
	Matière : Programmation Procédurale II Support : Prosit Séance : APP Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire : 2019-2020

Enoncé

On vous demande de développer une application pour une agence de voyages. Cette application permettra de gérer la liste des voyageurs et l'historique de leurs voyages en sauvegardant toutes les données dans un fichier **binaire**.

Un **voyageur** est caractérisé par :

- Un identifiant (chaîne de caractères)
- Un nom
- Une adresse
- Le nombre total de voyages

Pour ce faire, vous devez respecter les consignes suivantes :

- Lorsque l'agent de l'agence lance l'application, toutes les données existantes dans le fichier de sauvegarde doivent être chargées dans un tableau sur lequel des opérations d'ajout de suppression, de recherche ..., sont effectuées.
- Avant de quitter l'application, vous devez effectuer **une sauvegarde dans le fichier**.
- L'utilisation de la **mémoire doit être optimisée** (réservation selon le besoin et libération si on n'a pas besoin).

En respectant les consignes citées ci-dessus, Implémentez les fonctions suivantes :

1. **int nbVoyageurs(char nomFich[])** permettant de retourner le nombre de voyageur dans un fichier binaire.
2. **void saisirVoyageur (voyageur * ptv):** permettant de saisir les informations d'un voyageur.
3. **voyageur * charger_donnees (char nomFich[], int * n):** qui permet de:
 - remplir et retourner un tableau de voyageurs **tab** à partir du fichier donné par son nom.
 - retourner le nombre de voyageurs.

Remarque: Implémentez cette fonction avec 2 façons.(avec et sans l'appel de nbVoyageurs)

esprit Ecole Supérieure Privée d'Ingénierie et de Technologies	Objectif (2) : L'allocation dynamique	
	Matière : Programmation Procédurale II Support : Prosit Séance : APP Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire : 2019-2020

4. **void sauvegarder_donnees (char nomFich[], Voyageur * tab, int n)** qui permet de sauvegarder les voyageurs stockées dans le tableau **tab** dans un fichier donné par son nom.
5. **int trouver(voyageur * tab, char id[], int n)** qui permet de chercher un voyageur donné par son identifiant dans un tableau de voyageurs **tab** et retourner son indice s'il existe et -1 sinon.
6. **voyageur * ajouter (Voyageur v, voyageur * tab, int * n)** :qui permet d'ajouter le voyageur **v** dans un tableau de voyageurs **tab**. Cette fonction doit vérifier d'abord si le voyageur existe déjà : dans ce cas, le nombre de ses voyages est incrémenté. Sinon, il doit être ajouté à la fin du tableau.
7. **void supprimer (voyageur * tab , char id[], int * n)** : qui permet de supprimer le voyageur dont l'identifiant est passé en paramètre.
8. **void liberer (voyageur * tab)**qui permet de libérer l'espace mémoire du tableau de voyageurs **tab**.
9. Dans le programme principal, utilisez un menu permettant de tester les différentes fonctionnalités de votre application.

NB. Vous devez utiliser les fonctions prédéfinies malloc, realloc et free dans les fonctions adéquates !