

## Projet de TP : Gestion des Contacts « Fichier LOVC avec une Table d'Index »

### Enoncé

La gestion des contacts est une tâche importante au sein de nombreuses organisations. Elle permet de stocker des informations sur les clients, les prospects, les partenaires, etc. Cependant, la gestion des contacts peut s'avérer une tâche complexe, notamment si le nombre de contacts est important. Il est donc important de disposer d'un système efficace pour stocker et gérer les contacts, afin de pouvoir les retrouver facilement et de les mettre à jour rapidement.

Dans ce TP, on veut gérer les contacts dans un fichier binaire nommé '**Contacts.bin**'. Ce fichier est organisé selon la structure **LVC** (Organisation chaînée, format variable des enregistrements et avec chevauchement).

Chaque contact est caractérisé par les informations suivantes :

- **Identifiant unique (ID)** : Sur 8 caractères.
- **Nom complet** : Sur 30 caractères.
- **Numéro de téléphone** : Sur 10 caractères.
- **Adresse email** : Sur 30 caractères.
- **Observations** : Champ de **taille variable** pour des notes supplémentaires.

Les caractéristiques du fichier '**Contacts.bin**' sont stockées dans la structure '**entête**' et incluent les informations suivantes :

- Adresse du premier bloc du fichier.
- Taille du fichier.
- Nombre de contacts.
- Nombre de contacts insérés.
- Nombre de contacts supprimés.

### Partie I

Dans cette partie, le fichier n'est **pas ordonné** et les contacts sont stockés selon l'ordre de leur arrivée (**LVC**).

### Travail demandé

1. Définir et déclarer les différentes structures du fichier.

2. Déclarer le fichier et les variables globales.
3. Programmer les modules suivants :
  - 3.1. **Création** du fichier : Ce module consiste à remplir le fichier '**Contacts.bin**' avec N contacts (N est un entier positif qui doit être fourni par l'utilisateur). Afin d'éviter la saisie fastidieuse des données au clavier, le module doit automatiquement remplir le fichier avec des valeurs aléatoires comme suit :
 

**ID** : générer un nombre aléatoire compris entre 1 et N. Assurez-vous que l'ID est unique pour chaque contact (pas de doublons).

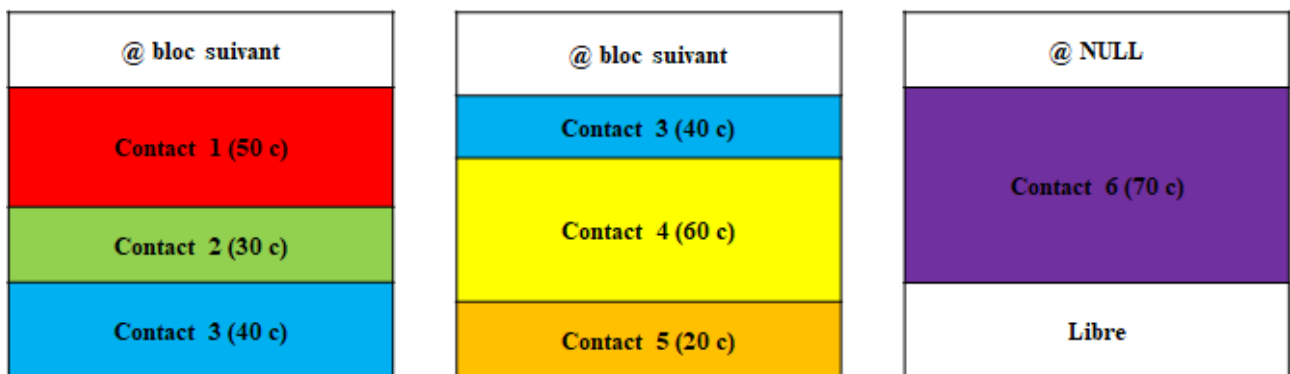
**Nom** : générer 30 caractères aléatoires.

**Numéro téléphone** : générer 10 nombres aléatoires chacun dans l'intervalle [0, 9].

**Adresse email** : générer 30 caractères aléatoires.

**Observations** : générer des caractères aléatoires. Le nombre de caractères est entre 50 et 250 caractères.
  - 3.2. **Recherche** d'un contact à partir de son ID : Ce module retourne l'adresse du bloc et le déplacement à l'intérieur de ce bloc où le contact est localisé, ou devrait être inséré (dans le cas d'un fichier ordonné).
  - 3.3. **Insertion** d'un nouveau contact donné par l'utilisateur.
  - 3.4. **Suppression logique** : Ce module permet de supprimer logiquement un contact (s'il existe) donné par son ID en modifiant l'indicateur d'effacement logique.
4. Après chaque **création**, **insertion** et **suppression**, il faudra afficher :
  - 4.1. Les caractéristiques du fichier de manière textuelle, sous forme d'un tableau descriptif.
  - 4.2. L'organisation globale et interne des blocs du fichier de manière graphique (voir schéma ci-dessous).

**Par exemple :**



Afin d'accélérer les opérations sur le fichier, nous **créons une table d'index dense** sur le champ **ID**. La table d'index est sauvegardée dans un **fichier séparé nommé 'Contacts\_Index.bin'**. La table d'index est **chargée** en MC **dès l'ouverture du fichier 'Contacts.bin'**.

Reprendre les questions ci-dessus tout en exploitant la table d'index. **La recherche d'un contact se fait par dichotomie dans la table d'index et se poursuit séquentiellement dans le bloc sélectionné.**

Une fois le fichier ‘**Contacts.bin**’ fermé, la table d’index mise à jour est enregistrée dans le fichier ‘**Contacts\_Index.bin**’.

## Partie II

Dans cette partie, on souhaite **créer un nouveau fichier** nommé ‘**Contacts\_Ordonnés.bin**’ à partir des données du fichier ‘**Contacts.bin**’. Dans ce fichier, les contacts sont **ordonnés** par ordre croissant selon le **champ ID**.

**NB :** le fichier ‘**Contacts\_Ordonnés.bin**’ ne doit pas contenir les contacts qui ont été **supprimés** dans le fichier ‘**Contacts.bin**’.

On crée aussi un fichier index nommé ‘**Contacts\_Ordonnés\_Index.bin**’ contient une **Table d’index non dense** sur le champ **ID**.

Reprendre les questions de la **Partie I**, en considérant la recherche **dichotomique** dans la **table d’index** et dans le **bloc sélectionné**.

## Ce qu’il faut rendre

- Un dossier zippé contenant le projet complet (codes sources + fichiers de données et d’index).
- Un rapport présentant le travail effectué :
  - Organisation du projet : description des différentes structures de données utilisées et les différents modules implémentés, un exemple de test de TP (captures d’écran), etc.
  - Difficultés rencontrées, les compétences nouvellement acquises, la répartition des tâches entre le binôme, la durée effective consacrée à la réalisation du projet, etc.

## Remarques importantes

- Le Projet TP est en **binôme**.
- Le TP doit être **opérationnel**.
- Les programmes doivent être implémentés en **langage C**.
- Les programmes doivent être bien commentés et bien organisés.
- La date de remise du projet est le **Jeudi 18 Janvier 2024**.

*Bonne chance.*