

## Compte Rendu TP C++ n°3 : Gestion des entrées/sorties

### I. Format du fichier démo :

Le fichier de chargement/sauvegarde est constitué d'une succession de lignes représentant soit un trajet simple soit un trajet composé telles qu'identifiées par le programme, et ce selon le format suivant :

#### **Format Trajet Simple :**

/villeDepart,villeArrivee,moyenTransport

Où l'étiquette / délimite le début d'un trajet simple.

, : sert de séparateur entre les données.

villeDepart : représente la ville de départ du trajet simple

villeArrivee : représente la ville d'arrivée du trajet simple

moyenTransport : représente le moyen de transport du trajet simple

#### **Format Trajet Composé :**

\$nombreTrajets,villeDepart,villeArrivee/villeDepart1,villeArrivee1,moyenTransport1/villeDepart2,villeArrivee2,moyenTransport2

Où l'étiquette \$ est caractéristique d'un trajet composé.

, : séparateur entre les données.

nombreTrajets : représente le nombre de trajets simples dont se compose le trajet composé.

villeDepart : représente la ville de départ du trajet composé, facilite la sélectivité lors du chargement des trajets.

villeArrivee : représente la ville d'arrivée du trajet composé, facilite la sélectivité lors du chargement des trajets tout en évitant le parcours du trajet pour retrouver la ville d'arrivée.

**N.B :** A la saisie, les villes de départ et d'arrivée des trajets simples consécutifs constituant le trajet composé sont demandées à l'utilisateur de sorte que seuls des trajets valides puissent être insérés dans le catalogue. Ceci permet d'assurer la cohérence du catalogue en réduisant les erreurs liées à l'utilisation. Cette abstraction n'est cependant pas visible au niveau du fichier de chargement/sauvegarde puisque celui-ci a été formaté de façon à privilégier des sous-formats réguliers afin de faciliter les entrées/sorties du programme tout en augmentant la réutilisabilité du code : un trajet composé n'étant composé que de trajets simples, ce choix de réalisation se répercute naturellement sur le format de sauvegarde.

### II. Contenu du fichier démo :

On réalise un fichier démo pour les trajets suivants :

Trajet Simple 1 : - ville de départ : Lyon  
- ville d'arrivée : Bordeaux  
- moyen de transport : Train

Trajet Simple 2 : - ville de départ : Lyon  
- ville d'arrivée : Paris  
- moyen de transport : Auto

Trajet Composé : 2 Trajets Simple

Trajet Simple 1 : - ville de départ : Lyon  
- ville d'arrivée : Marseille  
- moyen de transport : Bateau

Trajet Simple 2 : - ville de départ : Marseille  
- ville d'arrivée : Paris  
- moyen de transport : Avion

Ajoutons les trajets au catalogue puis sauvegardons celui-ci dans un fichier demo.txt.

**Contenu du fichier demo.txt :**

*/Lyon, Bordeaux, Train*

*\$2, Lyon, Paris/Lyon, Marseille, Bateau/Marseille, Paris, Avion*

*/Lyon, Paris, Auto*

### **III. Conclusion :**

L'implémentation initiale du programme nous a permis, comme prévu initialement, une souplesse lors de l'ajout de nouvelles fonctionnalités. En outre, aucune fuite mémoire n'a été déclarée suite au lancement de valgrind, comme lors du premier TP. Nous avons axé notre travail sur l'intuitivité et la fiabilité de l'interaction avec l'utilisateur et avons compris l'enjeu et l'importance des détails d'implémentation qu'il est nécessaire de prendre en considération dès la conception initiale pour faciliter l'évolutivité de l'application.

Parmi les axes d'évolution et d'amélioration fonctionnels possibles figure le développement de nouvelles fonctionnalités permettant un affichage graphique des différents trajets dans un graphe orienté et pondéré pour la visualisation du plus court chemin possible.