Compte-rendu de TP

TP n°1 C++ avancé : Entrées/sorties

Table des matières

I. Description du format de fichier……2
II. Axes d'amélioration et conclusion……..……..3

Note : Tapez "make dev" pour compiler avec -DMAP et voir les traces lors de l'exécution. Tapez "make help" pour plus d'informations sur les différentes commandes.

I. Description du format de fichier

Pour le format du fichier utilisé, nous avons décidé d'adopter une structure différente selon les trajets simples et les trajets composés. Dans les deux cas, nous avons utilisé le ";" en guise de séparateur entre chaque chaîne de caractère. Nous avions au départ pensé à un espace, mais le point-virgule s'est révélé plus pertinent pour éliminer le problème des villes qui auraient un nom contenant un espace.

a. Trajets simples

En ce qui concerne les trajets simples, nous avons décidé d'adopter le format suivant pour chaque trajet : "Ville de départ;Ville d'arrivée;Moyen de transport;". Lors du traitement, chacune de ces données est stockée sous forme de string dans un tableau. Un saut de ligne marque ensuite le passage au trajet suivant.

b. Trajets composés

Le début d'un trajet composé est marqué par la balise "COMPOUND". Pour les trajets simples, le passage au trajet suivant était simplement marqué par le saut de ligne après le dernier point-virgule, tandis que la description du trajet composé s'étend jusqu'à la balise "END_COMPOUND;". La première ligne prend la forme suivante : "COMPOUND;Ville de départ;Ville d'arrivée", où les villes de départ et d'arrivée sont celles du trajet composé en entier et non des villes intermédiaires. Après cette ligne, on décrit trajet par trajet la constitution du trajet composé en reprenant la syntaxe utilisée pour les trajets simples et composés. Une fois tous les trajets du trajet composé décrits, on marque la fin du trajet composé par la balise "END_COMPOUND;" à la manière d'une accolade fermante puis on effectue un saut de ligne pour passer au trajet suivant.

Ainsi, voilà ce que donne le contenu du fichier demo à tester demandé selon notre formatage :

```
Lyon;Bordeaux;Train;
Lyon;Paris;Auto;
COMPOUND;Lyon;Paris;
Lyon;Marseille;Bateau;
Marseille;Paris;Avion;
END_COMPOUND;
```

II. Axes d'amélioration et conclusion

Ce TP nous a permis de bien nous familiariser avec la manipulation des fichiers mais aussi des string, que nous avons cette fois-ci eu l'occasion d'utiliser.

L'une des difficultés rencontrées a été de trouver comment gérer les trajets composés pour faire le choix d'ajouter de nouveaux trajets en tant que partie de trajet composé ou en tant que nouveau trajet dans le catalogue. Nous avons pu y remédier à l'aide de variables booléennes indiquant si l'on est à l'intérieur d'un trajet composé ou non.

Nous avons aussi remarqué que selon la machine que l'on utilise, la fonction length() pour récupérer le nombre de caractères sur une ligne avait un fonctionnement différent. Sur les machines windows, il faut rajouter +1 à line.length() dans notre variable lengthLine pour que tout soit en ordre. Il aurait été bien de trouver une façon de faire qui ne nécessite pas cette légère modification selon la machine.

Une amélioration possible aurait été de mieux gérer les majuscules, notamment en ce qui concerne les outils de filtrage. Par exemple, si un trajet a pour départ "Lyon" dans le fichier démo et que l'on indique que l'on veut charger les trajets dont la ville de départ est "lyon", le trajet présent dans le fichier démo ne sera pas détecté. Avec plus de temps, nous aurions aimé corriger ce défaut.