Projet C - 2011

Contexte

Ce projet s'inscrit dans le cadre de données géo-localisées et de cartes, notamment avec http://www.world-gazetteer.com/ qui donne toutes les grandes villes de tous les pays avec leur population et les coordonnées GPS.

Travail demandé

Il s'agit de développer un program C qui implémente les fonctionnalités suivantes :

1. Lire les données d'entrée à partir d'un fichier CSV (2 points).

Le program doit prendre comme paramètre un fichier .csv qui contient une liste de villes données par leur nom, la population et les coordonnées géographiques (latitude et longitude) séparées par virgule. Chaque ville est sur une ligne comme dans l'exemple suivant :

```
Acheres,26482,48.97N,2.06E
Agde,25072,43.33N,3.46E
```

Les données sur les villes doivent être enregistrées dans une structure à votre choix.

2. Afficher les X plus grandes villes sur Google Maps (1 point).

Le program doit demander l'utilisateur un nombre X et afficher sur Google Maps les plus grandes X villes du fichier d'entrée. Pour faire l'affichage vous allez utiliser le fichier `map.html`, qui vous a été fourni, de la façon suivante :

- Vous allez lire le fichier `map.html` ligne par ligne
- Vous allez écrire un nouveau fichier `map-output.html`
- Vous allez remplacer les lignes concernant le tableau `coordinates` avec les coordonnées GPS des X premières villes
- (pour les points suivants vous allez remplacez aussi les lignes concernant le tableau `edges`)

3. Calculer le plus court chemin entre deux villes « en petit avion » (8 points).

Un *petit avion* est un avion qui peut voler seulement un nombre très limité de kilomètres (par exemple 300 km) avant de s'arrêter pour faire de nouveau le plein de kérosène. On considère qu'on à un aéroport dans toutes les villes du fichier d'entrée.

Le program doit demander à l'utilisateur de saisir les noms des deux villes. La première va être la source (ville de départ) et la deuxième va être la destination (ville d'arrivé). Il faut trouver le plus court chemin en petit avion de la source à la destination et l'afficher sur Google Maps.

Précisions :

- par le plus court chemin on entend la distance géographique à la surface de la Terre (on considère que les petits avions volent très bas). Donc le nombre de villes intermédiaires n'est pas important.
- pour calculer la distance entre deux points géographiques vous pouvez utiliser la formule qui est décrite ici http://www.movable-type.co.uk/scripts/latlong.html .
- pour calculer le plus court chemin vous allez utiliser les algorithmes que vous avez appris.
- **Bonus (1 point).** Pour choisir la ville de départ et d'arrivé le program va demander à l'utilisateur d'introduire 2-4 caractères et après il va afficher une liste avec toutes les villes qui commence avec les caractères donnés. L'utilisateur va choisir la ville en donnant sont numéro dans la liste, par exemple.
- **Bonus (2 points).** Implémenter un algorithme efficace.

4. Calculer l'arbre aérien couvrant minimal (4 points).
Pour un ensemble de N villes *l'arbre aérien couvrant minimal* c'est la façon de relier les N villes entre elles par N-1 lignes aériennes en petit avion de façon qu'on puisse aller de chaque ville à toutes les autres villes et que la somme des longueurs de toutes les lignes aériennes soit minimale.
Le program va demander le numéro N à l'utilisateur, il va calculer l'arbre aérien couvrant minimal pour les plus grandes N villes du fichier d'entrée et il va l'afficher sur Google Maps.

5. Lire des fichiers d'entrée dans un format plus complexe (2 points).
Ce point c'est à faire après vous allez apprendre ANTLR en compilation.

6. Analyse (3 points).
Vous allez analyser le code pour des fuites mémoire, vous allez analyser la performance en temps de calcul et vous allez estimer la complexité de votre solution.

La plateforme GitHub

Tout au long de ce projet vous allez utiliser la plateforme http://github.com pour votre code source. Chaque étudiant doit avoir un compte est doit faire partie d'une équipe.

Les instructions pour configurer Git se trouvent à l'adresse suivante :
https://github.com/polytech-projet-ig3/projet/wiki/Configuring-Git-and-Github

La page principale du projet est : https://github.com/polytech-projet-ig3

Chaque équipe à un répertoire Git sur github qui s'appelle « rX » où X e le numéro d'équipe.

Bonus (1 point). Utiliser toutes les fonctionnalités offertes par la plateforme GitHub : faire commit et push « assez souvent », le wiki, les tags, les commentaires sur les commits ou sur le code, des branches pour développer des différentes fonctionnalités, etc.

Evaluation

Le projet ne sera considéré comme fini qu'après validation sur un grand ensemble de tests (au moins 10) comprenant des tailles de fichiers d'entrées différents. Une **soutenance** aura lieu pendant laquelle vous devrez présenter tout ce que vous avez fait en 3 ateliers : présentation générale, livraison du code, présentation des résultats et tests effectués.

La version finale du projet va être marquée sur GitHub avec le tag « FINALE » et elle doit comporter :

- un *readme*
- un répertoire des fichiers sources (/src) avec
  - un *makefile*
  - un sous-répertoire des fichiers .c (/src/source)
  - un sous-répertoire des fichiers .h (/src/headers)
- un répertoire des tests et leurs résultats (/test)
- un répertoire de documentation comportant (/doc)
  - la charte de programmation adoptée
  - un rapport de synthèse (maximum 20 pages)
  - un rapport de tests
  - un schéma (PDF) de description des modules et de leurs interactions.

Le projet peut être développé en C ou C++ en suivant une charte de programmation que vous fournirez (un modèle pourra vous être fourni).