

## Réseaux : réseaux sociaux, métro, etc...

Le but du projet est de vous faire manipuler des réseaux. C'est un concept naturel : vous empruntez le réseau du métro, vous utilisez les réseaux sociaux, on peut avoir des réseaux de co-auteurs en recherche, etc. On pourra voir aussi ce projet comme une implémentation de concepts que vous avez vus dans votre cours de graphe.

Nous allons vous fournir deux fichiers en entrée. Le premier représente le réseau du métro parisien. Le second représente un réseau social qui a été généré aléatoirement. Le but du projet est d'écrire un code qui répond aux questions suivantes :

1. Ecrivez sur la console les premières 10 stations en ordre décroissant du nombre de stations directement voisines.
2. Ecrire une méthode qui prend en paramètre deux stations et qui retourne un itinéraire entre la première station et la seconde.
3. En utilisant les questions précédentes, implémentez des méthodes pour
  - (a) écrire sur la console le nom des dix personnes les plus connectées en ordre décroissant
  - (b) écrire la chaîne de rencontres nécessaires pour mettre en relation deux personnes passées en paramètre.
4. En utilisant la question 3b, implémentez une méthode pour savoir si l'on observe les "six degrés de séparation" (aussi connus comme les six degrés de Kevin Bacon).

Bonus. Trouvez un itinéraire qui utilise au moins un tronçon de chacune des lignes de métro en visitant le moins de stations possibles. C'est un problème difficile, donc je n'attends pas forcément une solution optimale, une "bonne" solution sera acceptable.

On vous demande également de rédiger un petit rapport (2 pages maximum)

- Une justification pour la structure de votre code.
- Une justification de votre méthode pour la question 4.
- Si vous faites la question bonus, expliquez votre algorithme et commentez vos résultats.
- Une discussion sur ce qui a été le plus dur à implémenter pour vous et sur le niveau de difficulté du projet.

## Soumission

1. Le projet est à rendre le **24 décembre 2018** sur la plateforme myCourse. Tout retard sera pénalisé.
2. Vous devez soumettre deux fichiers :
  - une archive zip dont le nom est la concaténation des noms des auteurs et qui contiendra l'ensemble des fichiers sources. Si les auteurs sont Astérix et Obélix, le fichier zip sera donc nommé `asterix_obelix.zip`. Lorsqu'on décompresse le fichier zip, on devra trouver un

répertoire nommé `asterix_obelix` contenant tous les fichiers sources. On devra pouvoir compiler et exécuter votre code depuis l'intérieur de ce répertoire.

- un document au format pdf contenant votre rapport nommé `asterix_obelix.pdf`. N'oubliez pas d'indiquer le nom des auteurs du travail dans le rapport.

Vous serez pénalisé si vous ne suivez pas ces consignes.

## Critères d'évaluation

Le projet est à effectuer de manière préférentielle en binôme. Si vous présentez un projet en trinôme, puisque votre force de travail augmente de 50%, la notation sera beaucoup plus sévère à travail égal.

L'évaluation portera sur la qualité de la structure de votre implémentation, sur la qualité de vos algorithmes (c'est un cours de programmation orientée objet), sur l'utilisation de ce qu'on a vu en cours, sur votre rapport, ainsi que sur une soutenance (qui sera organisée après la semaine d'examen). Cette soutenance ne demande aucune préparation de votre part. Elle durera une douzaine/quinzaine de minutes par groupe. La soutenance consistera en un échange au sujet de vos résultats, votre rapport, et du code. Si la soutenance fait apparaître qu'un des membres n'a pas beaucoup contribué, sa note pourra être revue à la baisse.

Ce projet compte pour 40% de la note de l'UE. Il est donc souhaitable que la note corresponde au travail de votre groupe, et non aux conseils d'autres groupes, d'autres étudiants ou d'internet. Si vous utilisez des sources (articles de recherche, posts sur internet, etc...), vous devez mentionner vos sources dans le rapport. On vous demande ici un exercice classique en informatique, donc il existe des solutions sur internet, mais nous pensons que c'est un très bon exercice pour votre formation. Si j'ai des soupçons de plagiat, je n'hésiterai pas à saisir le conseil de discipline.

## Fichiers en entrée

Les fichiers en entrée ont la structure suivante :

- une ligne commençant par `'%'` est un commentaire
- le fichier est partagé en deux sections : la première contient le nom de chaque noeud/sommet du réseau, la seconde contient tous les arcs/arêtes
- les deux parties du fichier sont séparées par une ligne contenant les caractères `###`.
- pour chaque noeud/sommet, chaque ligne contient le nom du noeud
- pour un arc/arête, les noms des deux stations sont séparés par
  - le caractère deux points `:` pour une arête (graphe non orienté). Par exemple  
`Porte Dauphine:Victor Hugo`  
`Victor Hugo:Charles de Gaulles - Etoile`
  - le caractère `>` pour un arc (graphe orienté). Par exemple `Danube>Botzaris`.