

## NOTICE INFORMATIVE

Pas plus de trois personnes ne peuvent choisir le même projet. Ces projets sont réalisés de manière strictement individuelle. Le but des projets est de réaliser des programmes en utilisant des méthodes classiques de bio-informatique.

Vous êtes fortement encouragé à contacter les personnes proposant les projets au plus tôt pour vous aider dans la compréhension de ceux-ci. Veuillez noter que vous devrez avoir une connaissance la plus approfondi possible des articles associés aux projets avant de prendre contact.

Vous pouvez discutez entre vous de vos projets, mais les codes seront comparés afin de vérifier qu'ils n'existent pas de similitudes entre eux. Il s'agit en effet d'un travail personnel qui vise à vous faire progresser de manière individuelle.

Vous avez normalement, pour un peu plus d'une semaine de travail complet.

Nous souhaitons également un petit rapport sur ce projet (2 pages maximum recto verso figures incluses pour le rapport scientifique + 4 pages maximum pour les annexes relatives à la programmation proprement dite).

Vous pourrez détailler en annexe les difficultés que vous avez eu. Pour les projets les plus « scientifiques », le rapport comportera une petite introduction sur le but du travail, le cadre théorique, le matériel et méthodes, les résultats et un petite discussion (similaire à un article scientifique). En annexe vous présenterez la structure des programmes réalisés. Vous présenterez également un exemple d'utilisation de chaque programme (ex : prog1.py fichier1 ).

Ce rapport vise à détailler votre démarche. Un schéma ou plusieurs schémas sont vivement conseillés afin de clarifier la mise en place de votre projet et des programmes.

Pour vous donnez une idée des éléments de notation de la programmation, voici une liste, non exhaustive, des différents critères :

- Toutes les questions du projet ont été traités
- Les programmes fonctionnent et produisent le résultat attendu
- Facilité d'utilisation des programmes
- Les programmes fonctionnent avec des arguments (ex nom de fichier, paramètres...) (modulé suggéré : argparse)
- Il y a une documentation associé au programme, fonction, classes ainsi que des commentaires dans le code (docstring, PEP8)
- Une fonction "help" a été implémentée pour chacun des programmes
- Concision du code

- Clarté du code
- Efficacité du code (rapidité d'exécution)
- Création et utilisation de modules
- Programmation orienté objet

Le rapport ainsi que les programmes devront "compiler" sous environnement Linux standard et devront contenir des fichiers exemples ainsi que des scénarios d'utilisations facile à mettre en place.

Une présentation individuelle présentant votre démarche et vos résultats aura lieu Vous préparez l'ordre de passage. La présentation sera de 7 minutes + 7 minutes de questions.

Costas Bouyioukos & Jean-Christophe Gelly