1ère année – Projet Informatique

DEVELOPPEMENT D'UN OUTIL DE GESTION
DE CANDIDATURE CENTRALISE

Objectif et pré requis

L'objectif de ce projet est d'initier une démarche projet de développement informatique mettant en œuvre une démarche d'ingénieur face à un projet relativement complexe. Il s'agit d'aborder le projet en évaluant les objectifs, les moyens dont vous disposez (compétences actuelles) et de gérer les risques inhérents à tout projet informatique (délais non respectés, verrous technologiques...).

Compétences requises : algorithmiques, Langage C (manipulation de tableau, tri, structure).

Environnement de programmation : Code::Block (www.codeblocks.org). Télécharger la version **nosetup**. (Menu Main > Downloads > Binaries : codeblocks-16.01**mingw**-nosetup.zip).

Evaluation de l'enseignement : Rapport de projet + Soutenance du projet.

Présentation du Cas d'Etude

Imaginons un groupe d'Ecoles d'Ingénieur souhaitant gérer de manière centralisée ses candidatures via un concours commun avec des règles homogènes. Les règles du concours sont les suivantes :

- Un candidat émet **5** vœux **classés** (de 1 à 5) parmi une liste de **10 Ecoles**.
- Chaque candidat se voit attribuer **1 seule note** de concours **valable pour toutes les écoles**. ***Corollaire*** : un candidat non éliminé est considéré comme classé dans tous ses vœux (tout ou rien).
- Chaque école propose un nombre de places offertes au concours.
- L'affectation dans une école tient compte du classement d'un candidat en regard du nombre de places offertes : si un candidat est classé 6ème dans une école proposant 5 places, il sera en liste d'attente (classé 1er en liste d'attente)
- Lorsqu'une place dans une école est offerte à un candidat, tous les vœux classés en dessous sont éliminés (et libèrent donc éventuellement une place aux autres candidats).
- Un candidat peut donc se retrouver dans l'une des situations suivantes :
    - il est admis dans son 1er vœu;
    - il est admis dans son n^{ème} vœux et placé en liste d'attente dans les vœux supérieurs;
    - il est en liste d'attente partout.

L'objectif est de proposer un système automatisé permettant de simuler des affectations en vue d'estimer par exemple la nécessité ou non d'affiner le nombre de candidats appelés en fonction de critères variés (taux de remplissage moyen par école, moyenne des candidats admis, satisfaction moyenne des candidats...).

Démarche Projet

Le projet présente un risque lié à sa complexité : comment gérer l'avancement du projet sans se retrouver avec beaucoup de fonctions incomplètes, *buggées*, avec une réponse incomplète et désordonnée aux objectifs. Afin de gérer ce risque, vous procéderez de manière incrémentale : c'est à dire que vous identifierez les fonctions indispensables et les fonctions "supports" c'est à dire améliorant l'exploitation du code et la présentation des résultats.

Grandes étapes du projet:

1. Compréhension du sujet, Expression des besoins

   ➔ Définir un jeu de données (candidats, écoles), simuler le processus d'orientation sur ce jeu de données, ...

2. Spécifications des grandes fonctions

   ➔ Avoir une vue très globale de l'enchainement des fonctions (reprenant les étapes réalisées manuellement dans la question 1.).

3. Implémentation et test d'UNE fonction (GOTO 2)

   ➔ Programmation en C, avec nom de variables intelligibles, commentaires du code source...

4. Finalisation d'une première version du logiciel (GOTO 1)

   ➔ test sur jeu de données, vérification & validation.

Afin de gérer progressivement la difficulté du projet, il vous est d'abord demandé de produire une **première version** du logiciel avec les hypothèses réduites suivantes :

- 6 candidats avec 3 vœux (tous remplis) parmi 4 écoles. Chaque école offre 2 places.
- Pas de recherche d'optimisation du code dans un premier temps
- Le programme devra contenir :
  - un jeu de données simple
  - définition des types de données
  - un tableau des candidats (avec liste de vœux, note au concours, état d'affectation)
  - un tableau des Ecoles (avec liste classées des candidats, nombre de places offertes)
  - une fonction de **tri par insertion** de la liste des candidats.
  - une fonction de remplissage du tableau des Ecoles à partir du tableau des candidats
  - une fonction utilisant les résultats précédents pour transcrire le résultat d'affectation des candidats (0 = admis, n > 0 = position en liste d'attente, -1 = vœux éliminé car vœu supérieur proposé)

Une fois cette première version obtenue, il sera possible de l'améliorer en ajoutant les fonctions suivantes :

- Optimisation si nécessaire du code.
- Extension du jeu de donnée pour tester la robustesse et les temps de calculs nécessaires.
- Lecture et sauvegarde des données dans un fichier texte.
- Ajout d'une interface de saisie pour les candidats, interface de consultation pour les écoles.

Le projet qui vous est soumis est de développer un outil permettant d'expérimenter le fonctionnement du nouveau concours avant la rédaction du cahier des charges pour produire un outil définitif. Aussi, le résultat de votre travail sera mesuré à la fois en terme de délivrables "binaire" que documentaire.

Travail à rendre

Le projet est organisé en 8 séances à l'issue desquelles vous devrez fournir :

- Le code source
- Un **document** présent la démarche suivie incluant notamment : le jeu de donnée utilisé pour tester votre code, l'organisation du travail, une analyse du travail accompli en incluant
    - une analyse des performances de calculs (imaginez ce que donnerait votre application avec un nombre de candidats plus grand)
    - les difficultés rencontrées, et avec le recul comment vous auriez abordé ce projet si vous aviez su tout ce que vous savez maintenant.
    - une analyse critique du fonctionnement du concours (sachant que tout fonctionnement a ses avantages et ses inconvénients).
- Une **présentation Powerpoint** qui résumera ce document. Attention : plus qu'un résumé, il s'agit de synthétiser votre travail, apprendre à le valoriser tout en apprenant à mettre en valeur quelques points clés... Contre exemple : commenter une page de code source.