

## Examen de reprise, Programmation2, INF1093, Été 2024

### Directives importantes :

1. L'examen est surveillé, aucune communication n'est autorisée
2. La documentation est autorisée mais exclut l'usage de l'IA générative
3. Durée : 2h
4. Créé un fichier python reprise.py qui contiendra tous tes scripts attendus
5. Ensuite, sou mets d'avance ce fichier via votre référentiel git. Aucun retard ne sera autorisé. Mieux vaut un fichier vierge qu'aucune soumission
6. Pour soumettre, exécute les commandes git suivantes à partir de la racine de ton référentiel git.
  - `git add .`
  - `git commit -m "numéro de la question"`
  - `git push`

Tu devras également faire la même chose après ta réponse à chaque question.

L'enseignant devra faire la ronde pour s'assurer de la bonne soumission des solutions tout au long de l'évaluation

### Contexte :

Considérons les deux listes suivantes :

- La première précise le matricule de chaque candidat
- La seconde indique la performance de chaque matricule.

Nous aimerons donner la liste indiquant le matricule, le nom et la performance de chaque candidat.

Matricule	Nom	Matricule	Notes/100
202401	Anani	202410	74
202402	Simon	202409	93
202403	Pierre	202408	59
202404	Kylian		
202405	Alphonse	202406	77
202406	Joshua		
202407	Colince	202404	90
202408	Kristian	202403	73
202409	Zinedine	202402	59
202410	Didier	202401	17

**Figure 1 : Données des candidats**

Pour le faire, deux approches seront adoptées :

1. L'une par recherche séquentielle
2. L'autre par recherche dichotomique

**Exercice 1 : Déclaration et initialisation des variables**

1. A base des valeurs indiquées dans la figure 1, déclarer, initialiser et afficher les 2 variables: **(5pts)**
  - i. **listIdGrade** qui contient la liste des couples (**matricule, note**)
    - a. Les matricules et les notes sont des valeurs numériques
  - ii. **fullList** qui contiendra la liste des triplets (**matricule, nom, note**) et mais est initialement une liste vide.

**listIdName** a été déclarée en guise d'exemple

**Exercice 2 : Recherche séquentielle** **(45pts)**

1. Écrire une fonction **searchIdByName(name)** qui met en œuvre la recherche séquentielle dans **listIdName** afin de renvoyer le matricule d'un élève à partir de son nom. **(12.5pts)**
  - a. Si le nom ne figure pas dans **listIdName**, elle renvoie 0.
  - b. Le parcours de la collection **listIdName** doit s'arrêter aussitôt que nous avons trouvé le nom l'élève. (IMPORTANT)
2. Écrire une fonction **searchGradeByIdSeq(id)** qui met en œuvre la recherche séquentielle dans la **listIdGrade** afin de trouver la note d'un élève à partir de son matricule **(12.5pts)**
  - a. Le parcours de la collection **listIdGrade** doit s'arrêter aussitôt que nous avons trouvé la note de l'élève. (IMPORTANT)
  - b. Si la note n'est pas trouvée, l'élève aura 0
3. Écrire la fonction **buildListSeq( )** qui élabore la **fullList** en réutilisant les fonctions implémentées dans les deux questions précédentes.
  - a. Si la note n'est pas trouvée, l'élève aura 0 **(15pts)**

**Exercice 3 : Tri et Recherche dichotomique** **(45 points)**

1. Trier la **listIdGrade** en fonction des matricules à l'aide d'un algorithme de tri de ton choix qu'il faudra préciser **(20 pts)**
2. Écrire une fonction **searchGradeByIdDicho(id)** qui met en œuvre la recherche dichotomique dans la **listIdGrade** triée afin de trouver la note d'un élève à partir de son matricule

3. Écrire la fonction **buildListDicho( )** qui élabore la **fullList** en réutilisant la fonction implémentée dans la question précédente. **(15pts)**
- a. Si la note n'est pas trouvée, l'élève aura 0
4. **Discussion** : Laquelle de ces deux approches te parait plus efficace ? Pourquoi? **(10pts)**