

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

**Structures de données et algorithmes**  
**Devoir 2 — Énoncé**

---

SIGLE : INF21307  
TITRE : Structures de données et algorithmes  
GROUPE : MS  
PROFESSEUR : Steven Pigeon  
K-212  
steven\_pigeon@uqar.ca

MODALITÉ : À faire seul ou en équipe de deux.  
DATE DE REMISE : 28 février 2023, avant minuit.

---

**Thème :** Tableau initialisé paresseusement.

Dans ce devoir, vous devez implémenter (langage au choix) un tableau initialisé paresseusement grâce à un *bitmap*.

**Total 15 points :** Langage au choix.

1. **3 pts.** Constructeur/destructeur. Votre objet doit prévoir une méthode pour l'initialisation qui prend en paramètre le nombre d'items dans le tableau. Le destructeur doit aussi faire le ménage correctement quand l'objet est détruit. Le *bitmap* doit être initialisé au début pour marquer qu'aucune des cases du tableau n'est occupée.
2. **3 pts.** Vous devez implémenter un accesseur équivalent à `[i]` pour accéder au  $i$ -ème élément.

Si vous y accédez en lecture, `[i]` doit retourner l'objet à la case  $i$  du tableau si la case est occupée et « vide » si la case est vide (selon le langage choisi, ça peut être `null`, une valeur réservée pour dire ceci-n'est-pas-un-objet ou encore un item initialisé par défaut). En écriture, `[i]` doit permettre d'aller écrire un item à la case  $i$  et mettre à jour le *bitmap* d'occupation.

3. **3 pts.** Vous devez implémenter « insérer à  $i$  », qui va insérer à la position  $i$  du tableau un item, en démenageant les autres items du tableau au besoin. Il faut bien entendu aussi mettre à jour le bitmap en conséquence !
4. **3 pts.** Vous devez implémenter « supprimer à  $i$  », qui va supprimer l'item à la position  $i$  du tableau, en démenageant les autres items du tableau au besoin et en mettant à jour le bitmap de la façon appropriée.
5. **1 pts.** Vous devez implémenter « libérer  $i$  », qui marque la case  $i$  comme étant maintenant libre.
6. **2 pts.** Vous devez fournir un programme pour démontrer que votre objet tableau paresseux fonctionne correctement. Vous devez y montrer que les insertions, suppressions et recherches fonctionnent comme on s'y attend.

*Indice* : Est-ce qu'insérer ou supprimer demande toujours de déménager des items ?

\*  
\* \*

Vous pouvez réutiliser des éléments du langage de programmation que vous choisirez. Si les objets *bitmap* et *array* (ou peu importe les noms qu'ils auront dans votre langage de programmation) existent, vous pouvez les réutiliser. Par contre, vous devez implémenter « tableau paresseux », même si quelque chose de similaire existe dans votre langage.

\*  
\* \*

La qualité du français écrit est de mise, et jusqu'à 10% de la note finale pourraient être retranchés. N'oubliez pas de citer vos sources, mêmes celles dont vous vous inspirez plus ou moins vaguement. Enfin, vous vous exposez à 20% de pénalité par jour de retard, jusqu'à concurrence de quatre jours — sauf circonstances *exceptionnelles*.

Pour la remise du devoir, préparez une archive (.zip) avec votre solution *au complet* mais *sans les fichiers temporaires ou intermédiaires*, et dont le nom sera `votrenom.devoir2.zip` (ou 7z ou ...). Cette archive sera déposée sur Moodle dans le dépôt prévu à cet effet.