

EXAMEN 2
[15% de la note finale]

CONSIGNES GÉNÉRALES

- La documentation personnelle, numérique ou physique, est autorisée. Aucun partage entre étudiants.
- À l'exception du site d'Omnivox, la navigation sur internet est interdite.
- Le plagiat, sous toutes ses formes, ou le partage de documents entraînent la note de zéro pour tout étudiant impliqué.
- Cet examen est d'une durée de 3 périodes.

CONSIGNES TECHNIQUES

- Créer un projet Java unique dans VS Code;
- Créer une classe Java différente pour chaque exercice avec une entête appropriée;
- Il n'est possible d'utiliser que les notions vues en classe. Il est interdit d'utiliser des classes externes autres que *Arrays*, *JOptionPane*, *Integer*, *Double*, *Math*, *System*, *DateFormat*, *Random*;
- Vous devez faire usage des bonnes pratiques de programmation, si applicables;
- Les messages de saisie doivent être le plus descriptifs possible;
- L'affichage produit doit être fidèle aux exemples fournis, s'il y a lieu;
- Effectuer les saisies via la méthode *JOptionPane.showInputDialog()*;
- Produire les affichages demandés en console ou via des boîtes de dialogue de la classe *JOptionPane*, selon les exemples fournis;
- Les commentaires dans le code sont facultatifs pour ce travail.

REMISE

- Remettre l'ensemble du projet dans une archive compressée dans la boîte de remise prévue à cet effet.
 - Toute remise effectuée après l'heure limite sera refusée. **PRÉVOYEZ VOTRE REMISE EN AVANCE.**
-

Exercice 1 – Somme maximale

[5 points]

À partir d'un tableau d'entiers de taille 15, remplir celui-ci de nombres générés aléatoirement dans l'intervalle inclusif [1, 20]. Votre programme doit identifier la somme maximale de 3 éléments adjacents du tableau et indiquer quelles positions ils occupent.

Éléments adjacents

Des éléments sont dits *adjacents* s'ils occupent des positions consécutives dans un tableau. Par exemple, les éléments aux positions 4, 5 et 6 sont adjacents, tout comme les positions 9, 10 et 11 le sont. Bref, **des éléments d'un tableau sont adjacents 3 à 3 si et seulement si leurs positions dans le tableau sont 3 entiers consécutifs.**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			adjacents						adjacents					

Exemples d’affichage du programme

Exemple 1	Dans [18, 14, 19, 9, 20, 7, 9, 19, 17, 9, 16, 18, 9, 17, 16] la somme maximale de 3 éléments adjacents vaut 51 (positions 1, 2 et 3)
Exemple 2	Dans [16, 1, 4, 1, 6, 6, 8, 3, 8, 2, 5, 14, 15, 13, 4] la somme maximale de 3 éléments adjacents vaut 42 (positions 12, 13 et 14)
Exemple 3	Dans [7, 8, 14, 9, 6, 2, 16, 12, 10, 9, 15, 12, 6, 11, 15] la somme maximale de 3 éléments adjacents vaut 38 (positions 7, 8 et 9)

Remarque : si plusieurs groupes de 3 éléments adjacents donnent la même somme maximale, il suffit d’afficher les positions d’un seul de ces groupes, celui de votre choix.

Exercice 2 – Pas encore des tableaux...

[10 points]

Écrire une mini-application qui manipulera un tableau de 10 entiers aléatoires compris dans l’intervalle inclusif [-10, 10]. Lors du lancement de votre application, le tableau devra être rempli une première fois, puis un menu principal affiche le tableau et propose les 4 options suivantes :



De plus, votre programme doit respecter les **exigences supplémentaires** suivantes :

- Dans votre code, on doit être en mesure de modifier à un endroit unique les 3 constantes suivantes et conserver le bon fonctionnement du programme tel que détaillé dans cet exercice :
 - la taille du tableau;
 - la limite inférieure de validité des nombres aléatoires;
 - la limite supérieure de validité des nombres aléatoires;
- Vous devez programmer chacune des fonctionnalités de l’application en réutilisant toujours le même tableau créé initialement, c’est-à-dire qu’il n’est pas permis de créer un second tableau.

1. **Regénérer les valeurs**

Un clic sur cette option va remplacer toutes les valeurs du tableau par de nouvelles valeurs aléatoires dans l’intervalle [-10, 10], puis on revient automatiquement au menu principal (qui affiche maintenant le nouveau tableau);

Exemple : Le tableau [8, -9, -5, 10, 2, -6, 6, -9, -3, -3] devient [-1, 0, -4, -3, 4, -1, -10, -2, 1, -8]

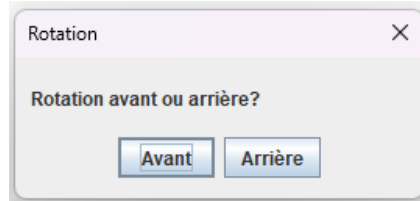
2. **Inverser**

Un clic sur cette option va « *inverser* » le tableau : la première valeur devient la dernière, la deuxième devient l'avant dernière, etc. Puis on revient au menu principal (en affichant maintenant le tableau inversé);

Exemple : le tableau [-1, 0, -4, -3, 4, -1, -10, -2, 1, -8] devient [-8, 1, -2, -10, -1, 4, -3, -4, 0, -1]

3. **Rotation**

Un clic sur cette option va proposer un choix à l'utilisateur : rotation *avant* ou *arrière* :



Avant

Si l'utilisateur clique sur le bouton « **Avant** », tous les éléments du tableau sont décalés d'une position vers l'avant (à droite) : la première valeur se retrouve en 2^e position, la deuxième valeur se retrouve en 3^e position, et ainsi de suite jusqu'à la dernière valeur qui se retrouve en 1^{ère} position.

Exemple : le tableau [-8, 1, -2, -10, -1, 4, -3, -4, 0, -1] devient [-1, -8, 1, -2, -10, -1, 4, -3, -4, 0]

Arrière

Si l'utilisateur clique sur le bouton « **Arrière** », tous les éléments du tableau sont décalés d'une position vers l'arrière (à gauche) : la dernière valeur se retrouve en avant-dernière position, l'avant-dernière valeur se retrouve en avant-avant-dernière position, et ainsi de suite jusqu'à la première valeur qui se retrouve en dernière position.

Exemple : le tableau [8, -9, -5, 10, 2, -6, 6, -9, -3, -3] devient [-9, -5, 10, 2, -6, 6, -9, -3, -3, 8]

Puis on revient au menu principal (en affichant maintenant le tableau modifié par rotation);

4. **Quitter**

Un clic sur cette option, ou sur le **X** en haut à droite du menu principal, doit faire quitter le programme correctement.

Bon succès!