

**Date de l'examen final** : Jeudi 12 décembre 2024 de 16h30 à 19h30 au Z-330 Pavillon Claire-McNicol

**Toute documentation papier (non électronique) est autorisée** : par exemple, les livres, notes de cours, codes des travaux pratiques imprimés sont permis. Tout comme l'examen intra.

**Le contenu et la matière du TP #3 font également partie de l'examen final.**

**Absence motivée aux examens** : <https://desi.umontreal.ca/reglements/>

Profitez des périodes de démonstration, Pratique Libre (PL) les samedis (9, 16, 23, 30 Nov et 7 décembre de 12 :30 à 16 :30 au S-118) pour avancer dans la réalisation de vos travaux pratiques à l'aide de vos auxiliaires d'enseignement.



Modalités :

À remettre, au plus tard, le **lundi 16 décembre 2024 à 23h59**

Le travail est corrigé sur **40 points**.

**Aucun travail remis en retard sera accepté.**

Les travaux se font seul ou en équipe de deux, au maximum.

Vous ne remettez alors qu'un seul travail.

**!/ ATTENTION !/** L'utilisation d'outils de génération de code par intelligence artificielle, et la copie de code d'une autre source, autre que ce cours IFT 1170, seront sévèrement **sanctionnées**.



Remise des travaux : même procédure que les TP #1 et #2.

### Numéro A (Vector : 16 points) :

Les matières du 7 et 14 novembre vous permettent de compléter ce numéro.

Voir les fichiers d'exemples **VectMetros.java** et **Vect2Metros.java** qui sont disponibles sur StudiUM.

Vous disposez du fichier texte nommé **pays\_a24.txt**. Chaque ligne contient les informations d'un seul pays. On a au maximum 225 pays.

Réalisez un programme en **Java** qui permet de réaliser les traitements ci-dessous :

1. Lire le fichier texte, remplir et retourner un vecteur (**Vector**) de pays.
2. Afficher seulement les 15 premiers pays lus.
3. Modifier le continent de la **Russie**, il devient un pays d'**Europe** ;  
Changer la population de l'**Allemagne** : cela devient 10 fois la population lue ;  
Supprimer le pays appelé « **DES OURAGANS** ».  
  
Nous réaffichons les 20 premiers pays du vecteur après ces modifications.
4. Trier selon les noms des pays, avec **Collections.Sort**, puis afficher les 10 premiers pays dans la console après le tri.
5. Chercher, en utilisant **Collections.BinarySearch**, puis afficher les pays suivants :  
**Canada, Mexique, Japon, Chili**

### Critères de correction :

Classe <b>Nation</b>	: 4 points
Bon fonctionnement avec les notions exigées	: 10 points (5 x 2 points)
Qualité de la programmation	: 2 points

### Numéro B : Liste des pays avec **LinkedList** (16 points) :

Les matières présentées à partir du 21 novembre vous permettent de compléter ce numéro.

Voir le fichier **Liste2.java** comme exemple sur StudiUM.

Vous disposez du fichier texte **pays\_bis.txt**.

Écrire un programme en **Java** permettant de réaliser les traitements ci-dessous :

1. Lire ce fichier texte, créer et retourner une liste (**LinkedList**) des pays d'**Amérique**.
2. Chercher séquentiellement, les pays dont leurs capitales sont respectivement les suivantes :  
**Washington, Ottawa, Santiago**

3. Trier, en utilisant **Collections.Sort**, la liste selon les capitales des pays.  
Afficher dans la console, seulement les 7 premiers pays de cette liste.
4. Ajouter le pays suivant, en conservant la liste triée selon les capitales :
- |              |                 |               |                 |
|--------------|-----------------|---------------|-----------------|
| <b>CHILI</b> | <b>SANTIAGO</b> | <b>756950</b> | <b>15328467</b> |
|--------------|-----------------|---------------|-----------------|

Finalement afficher dans la console, tous les pays de la liste.

**Critères de correction :**

Classe <b>Nation</b>	: 5 points
Bon fonctionnement avec les notions exigées	: 8 points (4 x 2 points)
Qualité de la programmation	: 3 points

**Numéro C (itération et récursivité : 8 points) : matière du 28 novembre**

Dans un premier temps, écrivez une version itérative pour afficher verticalement un entier **n > 0**.  
Par la suite, écrivez une version récursive, pour afficher la même chose dans la console.

Par exemple pour **n = 2731**, nous afficherons dans la console :

**2**  
**7**  
**3**  
**1**

**Conseil** pour la version itérative : convertir **n** en **String** puis afficher la chaîne verticalement.

Données pour la remise : **n = 2731** et **n = 41376**.

**Critères de correction :**

- Version récursive : 4 points
- Version itérative : 4 points

**Bonne fin de session, bonnes vacances et bonnes fêtes de fin d'année !**  
**Équipe du IFT 1170, Automne 2024**

---

**Durant la session d'Hiver 2025 :**

Le cours, IFT 1166 : Programmation orientée objet en C++, sera enseigné par Van Nguyen Le.

L'horaire sera ci-contre :

**Cours théorique** : Mercredi de 18h30 à 20h30, du 8 janvier au 9 avril 2025

**Séances de démonstrations** :

A101 : Mercredi de 16h30 à 18h30, du 15 janvier au 9 avril 2025

A102 : Mercredi de 20h30 à 22h30, du 15 janvier au 9 avril 2025

Pour plus d'informations : <https://admission.umontreal.ca/cours-et-horaires/cours/ift-1166/>

Le C++ est un langage de programmation très populaire dans l'industrie.

À l'issue de ce cours, vous connaîtrez les bases de la programmation orientée objet en C++, qui est un ensemble de concepts extrêmement important, et que nous retrouvons également dans d'autres langages. Nous vous en parlerons davantage durant les séances de démonstrations.