

TP2

○ Objectifs du travail pratique :

- Appliquer les piles, files et listes chaînées dans des scénarios d'applications réels.
- Approfondir la compréhension des files circulaires et des files à priorité dans des situations dynamiques.
- Manipuler des listes doublement chaînées dans des contextes nécessitant des parcours dans les deux directions.

Exercice 1 : Gestion des Annulations d'Actions (Système de Gestion d'Actions)

- Implémenter un système de gestion d'actions pour un éditeur de texte en utilisant une pile.
- Chaque action effectuée (par exemple : ajouter, supprimer ou modifier un texte) doit être enregistrée dans une pile.
- Implémenter la fonctionnalité "Annuler" qui permet de revenir à l'action précédente.
- Ajouter une fonctionnalité "Restaurer" qui permet de réappliquer une action annulée.
- **Tests** : Créer un cas où plusieurs actions sont effectuées, puis tester les fonctionnalités d'annulation et de restauration.

Exercice 2 : Simulation d'une File de Sécurité Aéroportuaire (Gestion des Passagers)

- Simuler une file d'attente de passagers pour un contrôle de sécurité en utilisant une file circulaire.
- Chaque passager peut être marqué comme "VIP" ou "Standard".
- Les passagers VIP doivent être ajoutés directement en tête de file (en contournant les autres).
- Implémenter les opérations ajouter_passager et servir_passager, où un passager est servi en fonction de l'ordre d'arrivée ou de la priorité "VIP".
- **Tests** : Simuler l'ajout et la gestion de plusieurs passagers et vérifier que les VIP sont bien servis en priorité.

Exercice 3 : Gestion d'une File à Priorité pour une Application de Messagerie (File d'Envoi de Messages)

- Créer une file à priorité pour gérer l'envoi de messages dans une application de messagerie.
- Chaque message a une priorité (par exemple : "Haute", "Moyenne", "Basse").
- Les messages avec une priorité plus élevée doivent être envoyés avant les autres.
- Implémenter les opérations `envoyer_message`, `ajouter_message` et `consulter_prochain_message`.
- **Tests** : Ajouter plusieurs messages de différentes priorités et vérifier l'ordre d'envoi.

Exercice 4 : Gestion d'une Playlist de chansons (Système de Playlist)

- Implémenter une playlist de chansons où chaque morceau de chanson est un nœud d'une liste doublement chaînée circulaire.
- Permettre les opérations suivantes : `ajouter_morceau`, `supprimer_morceau`, `suivant`, et `précédent`.
- L'utilisateur doit pouvoir naviguer entre les morceaux en avant et en arrière.
- **Extension** : Ajouter une fonctionnalité de "Lecture Aléatoire" qui permet de sélectionner un morceau au hasard.
- **Tests** : Créer une playlist avec plusieurs morceaux et tester les fonctionnalités de navigation et de suppression.
- **Évaluation**
 - Implémentation correcte des structures de données.
 - Adaptation des structures pour répondre aux exigences des scénarios.
 - Tests complets et vérification de la gestion des cas limites.