

# forward\_list (séance encadrée tp311)

Pour cette séance

- créer un projet **M3103 - MyForwardList** dans NetBeans (cf. Aide NetBeans).
- copier les fichiers à compléter disponibles dans /users/info/pub/2a/M3103/tp311 dans le répertoire créé par NetBeans (cf. Aide NetBeans).
- importer les fichiers dont l'extension est `.h` ou `.cpp` (cf. Aide NetBeans).

## Exercice 1 : Premier contact avec la classe `forward_list`

Créer une configuration **PremierContact** dans le projet **M3103 - MyForwardList** avec comme point d'entrée le fichier `premiercontact.cpp`.

Compléter le fichier `premiercontact.cpp` afin de produire une trace d'exécution la plus proche possible de celle qui est proposée et observer la fonction `getInfoAtPosit()` ainsi que la procédure `insereInfoAtPosit()`.

## Exercice 2 : i-ème entier pair

Créer une configuration **AutourDePair** dans le projet **M3103 - MyForwardList** avec comme point d'entrée le fichier `autourdepair.cpp`.

Compléter le fichier `autourdepair.cpp` avec les fonctions suivantes :

- `forward_list<int>::iterator premierEntierPair(forward_list<int>& uneListe)` qui permet d'accéder au premier entier pair d'une liste d'entiers ;
- `forward_list<int>::iterator iemeEntierPair(forward_list<int>& uneListe, int i)` qui permet d'accéder au i<sup>ème</sup> entier pair d'une liste d'entiers.

Et tester les fonctions implantées.

## Exercice 3 : Une `forward_list` est-elle triée ?

Créer une configuration **AutourDuTri** dans le projet **M3103 - MyForwardList** avec comme point d'entrée le fichier `autourdutri.cpp`.

Compléter le fichier `autourdutri.cpp` en implantant le prédicat `bool estTriee(forward_list<T>& uneListe)` qui vérifie qu'une `forward_list` est triée, essayer votre méthode sur des listes contenant des entiers, des flottants et des chaînes.

## Exercice 4 : Une liste sans élément dupliqué

Compléter le fichier `autourdutri.cpp` en implantant la fonction `forward_list<T> nouvelleListeTrieSansDuplication(forward_list<T>& uneListe)` qui produit une liste sans élément dupliqué à partir d'une liste triée qui peut en contenir.

## Pour cette séance

- créer un projet **M3103 - MySet** dans NetBeans (cf. Aide NetBeans).
- copier les fichiers à compléter disponibles dans /users/info/pub/2a/M3103/**tp312** dans le répertoire créé par NetBeans (cf. Aide NetBeans).

## Exercice 5 : Premier contact avec la classe set

Compléter le fichier `premiercontact.cpp` afin de produire une trace d'exécution la plus proche possible de celle qui est proposée.

Il faut pour cela compléter successivement les procédures et fonctions suivantes :

- ```
- void testeCopieEnsemble(),
- TypeElem getElemAtPosition(set<TypeElem> monEnsemble,
                             int laPosition),
- int getPositionDansEnsemble(set<TypeElem> monEnsemble,
                              TypeElem monElement),
- bool supprimeElement(set<TypeElem>& monEnsemble,
                       TypeElem monElement),
- void testeEnsembleString(set<string> lEnsemble),
- void testeOperationsEnsemblistes(),
```

et les appels aux procédures de teste dans la procédure **main()** à savoir, l'appel à :

- ```
- void testeCopieEnsemble(),
- void testeEnsembleString(set<string> lEnsemble),
- void testeOperationsEnsemblistes().
```