TD10 : éléments de correction

Exercice 1

1. Construire une liste d'entiers contenant 3, 4, 2, 6, 12, 15, 9, 5, 2 directement

Soit Lint = liste d'entiers initialisée à {3, 4, 2, 6, 12, 15, 9, 5, 2}

1. Construire une liste d'entiers avec une boucle de saisie

```
Soit Lint = liste d'entiers
Pour i de 0 à 7 avec un pas de +1 Faire
|       Soit n = SaisirEntier()
|       Ajouter(Lint, n)
Fin Pour
```

```
Fonction : CreerListeFibonacci
Entrée :        - n : entier
Local :         /
Sortie :        - fibo : liste d'entiers
Début
|       Soit fibo = liste d'entiers
|       Si n < 0 Alors
|       |       Retourner fibo
|       Fin Si
|       Si n == 0 Alors
|       |       Ajouter(fibo, 0)
|       |       Retourner fibo
|       Fin Si
|       Ajouter(fibo, 0)
|       Ajouter(fibo, 1)
|       Pour i de 2 à n avec un pas de +1 Faire
|       |       Ajouter(fibo, fibo[i − 1] + fibo[i − 2])
|       Fin Pour
|       Retourner fibo
Fin
```

```
Fonction : RechercheIndicePremiereApparitionTrie
Entrée :        - liste : liste d'entiers
                - n : entier
Local :         /
Sortie :        - entier
Début
|       Pour i de 0 à Longueur(liste) − 1 avec un pas de +1 Faire
|       |       Si liste[i] == n Alors
|       |       |       Retourner i
|       |       | // on peut s'arrêter dès qu'une valeur supérieure à n est rencontrée car la liste est triée
|       |       Sinon Si liste[i] > n Alors
|       |       |       Retourner -1
|       |       Fin Si
|       Fin Pour
|       Retourner -1
Fin
```

```
Fonction : RechercheIndicePremiereApparitionNonTrie
Entrée :        - liste : liste d'entiers
                - n : entier
Local :         /
Sortie :        - entier
Début
|       Pour i de 0 à Longueur(liste) − 1 avec un pas de +1 Faire
|       |       Si liste[i] == n Alors
|       |       |       Retourner i
|       |       Fin Si
|       Fin Pour
|       Retourner -1
Fin
```

```
Fonction : RecherchePlusGrandeMonotonieCroissante
Entrée :        - liste : liste d'entiers
Local :         - debut, indice, taille, tailleMax : entiers
Sortie :        - monotonie : liste d'entiers
Début
|       Soit monotonie = liste d'entiers
|       Si Longueur(liste) == 0 Alors
|       |       Retourner monotonie
|       Fin Si
|       Soit debut = 0
|       Soit indice = 1
|       Soit taille = 1
|       Soit tailleMax = 1
|       Tant Que debut < Longueur(liste) Faire
|       |       Soit indice = debut + 1
|       |       Soit taille = 1
|       |       Tant Que indice < Longueur(liste) ET liste[indice] ≥ liste[indice − 1] Faire
|       |       |       Soit indice = indice + 1
|       |       |       Soit taille = taille + 1
|       |       Fin Tant Que
|       |       Si taille > tailleMax Alors
|       |       |       // copie 'taille' éléments de 'liste' depuis l'indice 'debut'
|       |       |       Soit monotonie = CopierListe(liste, debut, taille)
|       |       Fin Si
|       |       Soit début = debut + taille
|       Fin Tant Que
|       Retourner monotonie
Fin
```