

Exercice 5

// 1. Déclarer en C des types MAILLON et PTR permettant de réaliser des vecteurs creux.

```
class Maillon {  
    int index;  
    float valeur;  
    Maillon suivant;  
}
```

// 2. Ecrire une fonction nouveau_maillon permettant de créer un nouveau maillon avec des valeurs initiales fournies en arguments.

```
public static Maillon nouveauMaillon(int index, float valeur) {  
    return new Maillon(index, valeur);  
}
```

// 3. Ecrire une fonction PTR vecteur_creux (float t[], int n) qui prend un tableau t ayant n éléments et construit sa représentation sous forme de vecteur creux.

```
public static Maillon vecteurCreux(float[] t, int n) {  
    Maillon tete = null, courant = null;  
  
    for (int i = 0; i < n; i++) {  
        if (t[i] != 0) {  
            Maillon nouveau = new Maillon();  
            nouveau.index = i;  
            nouveau.valeur = t[i];  
            nouveau.suivant = null;  
  
            if (tete == null) {  
                tete = nouveau;  
                courant = nouveau;  
            } else {  
                courant.suivant = nouveau;  
                courant = nouveau;  
            }  
        }  
    }  
    return tete;  
}
```

// 4. Ecrire une fonction PTR somme (PTR a, PTR b) qui reçoit deux vecteurs creux a et b et
// retourne le vecteur creux qui représente leur somme (c'est-à-dire l'addition des deux
// vecteurs, composante par composante).

```
public static Maillon somme(Maillon a, Maillon b) {  
    Maillon resultat = null;  
    Maillon courant = null;  
  
    while (a != null || b != null) {  
        Maillon nouveau = null;  
  
        if (a != null && (b == null || a.index < b.index)) {  
            nouveau = nouveauMaillon(a.index, a.valeur);  
            a = a.suivant;  
        } else if (b != null && (a == null || b.index < a.index)) {  
            nouveau = nouveauMaillon(b.index, b.valeur);  
            b = b.suivant;  
        } else if (a != null && b != null && a.index == b.index) {  
            float somme = a.valeur + b.valeur;  
            if (somme != 0) {  
                nouveau = nouveauMaillon(a.index, somme);  
            }  
            a = a.suivant;  
            b = b.suivant;  
        }  
  
        if (nouveau != null) {  
            if (resultat == null) {  
                resultat = nouveau;  
                courant = nouveau;  
            } else {  
                courant.suivant = nouveau;  
                courant = nouveau;  
            }  
        }  
    }  
    return resultat;  
}
```