

Exercice 3 : Listes chaînées circulaires

On considère le type `struct liste_chaine_circulaire` défini par :

```
struct liste_chaine_circulaire {  
    int valeur;  
    struct liste_chaine_circulaire *suivant;  
};
```

Le champ `suivant` du dernier élément d'une liste de type `liste_chaine_circulaire` pointe sur le premier élément de la liste.

1. Ecrire une fonction qui prend en arguments une valeur entière et une liste chaînée circulaire et renvoie le nombre d'occurrences de la valeur dans la liste.
2. Ecrire une fonction qui prend en arguments deux listes chaînées circulaires et les concatène, sans création de nouvelle liste.
3. Ecrire une fonction qui prend en arguments deux listes chaînées circulaires et crée une nouvelle liste qui est une concaténation des deux précédentes.
4. Ecrire une fonction qui prend en arguments une valeur entière et une liste chaînée circulaire et supprime de cette dernière toutes les occurrences de la valeur transmise (on évitera de laisser traîner en mémoire des données auxquelles on n'a plus accès).