

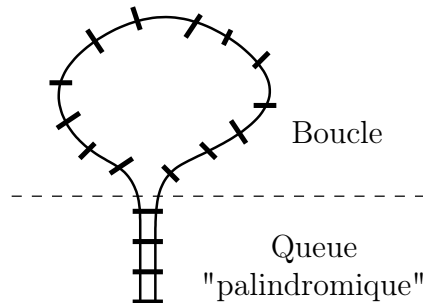
Chaînes de caractères et palindromes

Exercice 1 : Écrivez une fonction `inverse` qui prend en entrée une chaîne de caractères et fournit en sortie la chaîne inversée. Par exemple `inverse ("Gaston")` fournit la chaîne `"notsaG"` en résultat.

Exercice 2 : Écrivez une fonction `palindrome` qui prend en entrée une chaîne de caractères et fournit en sortie un booléen qui dit si c'est un palindrome.

- Faites une première version simple qui consiste à comparer la chaîne avec son inverse (calculé grâce à la fonction `inverse` de l'exercice précédent).
- La première version a le désavantage de parcourir 2 fois la chaîne de caractères (une fois pour l'inverser et une fois pour la comparer avec l'original). Écrivez une seconde version, 4 fois plus rapide, qui se limite à deux fois moins de comparaisons que de caractères dans la chaîne.

Exercice 3 : Écrivez une fonction `boucle` qui prend en entrée une séquence d'ARN et qui fournit en sortie la boucle maximale obtenue en supprimant aux extrémités les sous-séquences qui se replient l'une sur l'autre selon le dessin suivant :



Exercice 4 : Écrivez une fonction `phase` :

- qui prend en entrée deux chaînes de caractères `c` et `b`, où `c` est supposé être un codon (donc chaîne d'ATGC de 3 caractères exactement) et `b` un brin d'ADN (donc une chaîne ne contenant que des ATGC),
- qui fournit en sortie la phase de lecture (1, 2 ou 3) du codon dans le brin,
- et 0 dans le cas où le codon n'est pas dans le brin.

Exercice 5 : (*exercice subsidiaire*)

Écrivez une fonction `boucle2` qui étend la fonction `boucle` de l'exercice 3 en acceptant que l'une des deux extrémités du brin d'ARN « dépasse » de la queue palindromique.

Encore mieux (`boucle3`) : ne vous arrêtez pas au premier palindrome rencontré, ne retenez que le plus long d'entre eux. . .

Rappel : chacune de vos fonctions peut faire appel à des fonctions que vous avez déjà programmées auparavant.