

TP4 – Les boucles (suite)

Exercice 1 -

Ecrire un programme permettant à un utilisateur de saisir une séquence d'ADN et de la transformer en son « inverse complémentaire ». Le programme affichera la séquence inverse-complémentaire dans les sens 5' --->3'.

Note : proposer deux solutions différentes à cet exercice !

Exercice 2 -

Ecrire un programme permettant à un utilisateur :

- de saisir une séquence d'ADN,
- de vérifier si la séquence entrée est bien une séquence d'ADN (cf. Aide ci-dessous),
- d'afficher le nombre d'erreurs ainsi que la position de chacune des erreurs.

Aide : on a vu que l'opérateur « in » permettait de tester si une chaîne est (ou n'est pas) présente dans une autre chaîne ; l'opérateur « not in » permet quant à lui de tester si une chaîne n'est pas (ou est) présente dans une autre chaîne.

Exemple : "F" not in "ATGC" renvoie « VRAI » (true). On utilisera cet opérateur avec profit pour traiter la question considérée.

Exercice 3 -

Revoir le programme précédent, en sortant de la boucle dès qu'une erreur est détectée.

Note : utilisation d'un flag.

Exercice 4 -

Ecrire un programme permettant à un utilisateur :

- de saisir une chaîne de caractères,
- et de vérifier s'il s'agit d'un palindrome.

Exemple : l'utilisateur entre la chaîne "LAVAL" et le programme affiche « il s'agit d'un palindrome ».

Exercice 5 -

Ecrire un programme permettant à un utilisateur :

- de saisir une séquence d'ADN,
- et de vérifier s'il s'agit d'un palindrome « biologique » type épingle à cheveux.

Exemple : l'utilisateur entre la chaîne "TACGTA" et le programme affiche « il s'agit d'un palindrome ».

5'T A 3'
A T
C G