TP3 – Les boucles

Introduction

Les boucles sont un des fondements de la programmation. Elles permettent d'itérer (de répéter) un certain nombre de fois un groupe d'instructions. Il existe **deux types de boucle**, les boucles "for" et les boucles "while". On ne traitera dans ce TP que les boucles "while".

Ce TP va nous permettre également d'introduire la notion de **variable d'accumulation** (cette notion est indépendante du langage de programmation utilisé). Ce type de variable permet de construire (dans le cas de la concaténation) ou de calculer (dans le cas numérique) incrémentalement une chaîne ou un calcul.

Exemple de variable d'accumulation avec des entiers :
la somme des 3 premiers entiers peut s'écrire :

```
a = 1
a = a + 2
a = a + 3
print a          # a vaut 6
```

Rq1 : ici au lieu de l'instruction a = a + 3 on pourrait écrire a = 3 + a sans changer le résultat.

Exemple de variable d'accumulation avec des chaînes de caractères : concaténation

```
a = "Bonjour"
a = a + " Robert"
print a          # a vaut Bonjour Robert
```

Rq2 : contrairement au cas précédent écrire a = "Robert " + a change le résultat qui serait dans ce cas: Robert Bonjour

On verra, lors de l'exercice 5, que la notion de variable d'accumulation prend tout son sens lorsqu'elle est combinée avec une boucle.

Exercices

Exercice 1 -
Ecrire un programme permettant à un utilisateur de saisir une séquence d'ADN et d'afficher séparément chaque caractère de la séquence. Et dans un deuxième temps, on affichera tous les caractères sur la même ligne.
Note : une virgule en fin de commande print, vous évite de passer à la ligne.

Exercice 2 -
Ecrire un programme permettant à un utilisateur de saisir une séquence d'ADN et d'afficher chaque caractère suivi de sa position dans la séquence à l'écran.
Exemple : si la chaîne saisie par l'utilisateur vaut « ATGC » votre programme doit afficher :

```
A est en position 1
T est en position 2
G est en position 3
C est en position 4
```

Rq : Nous rappelons que la position est différente de l'index (cf. TP1).

Exercice 3 -
Ecrire un programme permettant à un utilisateur de saisir une séquence protéique et d'afficher le nombre d'acides aminés soufrés (sans utiliser la méthode « count »).
ex : Si la séquence entrée est : MATYLVKMCPGCRRAIL, alors le programme affiche « 4 acides aminés soufrés ».

Exercice 4 -
Ecrire un programme permettant à un utilisateur de saisir une séquence d'ADN puis d'afficher le pourcentage en A, T, G et C de la séquence (sans utiliser la méthode « count »).

Exercice 5 -
Ecrire un programme permettant à un utilisateur de saisir une séquence d'ADN puis de générer, en utilisant une **variable d'accumulation,** une nouvelle séquence correspondant à la transcription de cette séquence d'ADN.

Exemples de QCU python
Chaque question a une réponse unique.
Chaque réponse à une question sera notée de la façon suivante :

- **une réponse juste = +1 point**
- **pas de réponse ou "je ne veux pas répondre à la question" = 0 point**
- **une réponse fausse = -1/2 point**

```
seq='aTGc '
seq1=seq.capitalize()
print seq1
```

Qu'affiche le programme?
☐ aTGc
☐ AtgC
☐ Atgc
☐ Je ne veux pas répondre à la question

```
5!=3
```

Que retourne Python ?
☐ true
☐ false
☐ error
☐ Je ne veux pas répondre à la question

```
a=3
a=a+3
```

Quel est le contenu de la variable « a » ?
☐ 3
☐ la deuxième instruction ne veut rien dire
☐ 6
☐ Je ne veux pas répondre à la question

```
a=10
if a % 5 ==0 :
    print "OK",
print "..."
```

Qu'affiche le programme?
☐ OK ...
☐ OK
☐ ...
☐ Je ne veux pas répondre à la question