

## Examen Médian

Durée : 1 heure 30

Document autorisé : une feuille de notes A4 recto-verso  
Calculatrices, téléphones, traducteurs et ordinateurs interdits.

**Attention : chaque partie doit être rédigée sur une copie séparée**

N.B. : on s'attachera à fournir tout commentaire utile et à écrire de façon claire et lisible.

### 1<sup>ère</sup> partie : alternatives (6 points)

1. Écrire un programme permettant de saisir le prix de fabrication et le prix de vente d'un produit et de déterminer s'il y a profit ou perte.  
Si le prix de fabrication est supérieur au prix de vente, il y a perte et sinon profit.  
Améliorer ce programme pour permettre à l'utilisateur de saisir plusieurs prix de produits. La saisie s'arrêtera lorsque l'utilisateur entrera 0 pour le prix de fabrication.
2. Écrire un programme permettant d'afficher le nombre de jours d'un mois en fonction de son numéro (entre 1 et 12).

N.B. : si le numéro de mois est 2 (mois de février), il faudra demander à l'utilisateur le numéro de l'année afin de déterminer si l'année est bissextile et préciser alors 28 ou 29 jours.

On rappelle qu'une année est bissextile si elle est divisible par 4. Toutefois, les années divisibles par 100 ne sont pas bissextiles, à l'exception des années divisibles par 400 qui, elles, le sont.

Le nombre total de jours dans un mois est donné par le tableau ci-dessous.

Mois	Nombre de jours
janvier, mars, mai, juillet, août, octobre, décembre	31 jours
février	28/29 jours
avril, juin, septembre, novembre	30 jours

----- { prendre une nouvelle copie } -----

### 2<sup>ème</sup> partie : séquences d'ADN (7 points)

Une séquence d'ADN (acide désoxyribonucléique) est constituée de 4 éléments : Adénine (A), Cytosine (C), Guanine (G) et Thymine (T).

On appellera « chaîne » d'ADN une chaîne de caractères constituée de caractères A, C, G ou T, et « motif » une sous-chaîne d'une chaîne d'ADN.

Une chaîne ou un motif sont « valides » s'ils ne sont pas vides et s'ils sont formés exclusivement d'une combinaison arbitraire de 'A', 'T', 'G' ou 'C'.

- 1- Écrire un algorithme pour une fonction `valide(ch)` qui renvoie `vrai` si la chaîne `ch` passée en paramètre est valide et `faux` sinon. Écrire ensuite la fonction Python correspondante.
- 2- Écrire une fonction python `compte(ch)` qui renvoie le nombre de caractères de la chaîne `ch` si elle est valide et `-1` sinon.
- 3- Écrire une fonction `saisie()` qui effectue une saisie valide et renvoie la valeur saisie sous forme d'une chaîne de caractères.
- 4- Écrire une fonction `proportion(motif, ch)` qui reçoit deux arguments, un motif et une chaîne et qui retourne la proportion d'occurrences du motif dans la chaîne (c'est-à-dire le nombre de fois où le motif apparaît dans la chaîne).
- 5- Écrire le programme principal qui appelle la fonction `saisie()` pour la chaîne et pour le motif et qui affiche un résultat analogue à l'exemple suivant :

```
Entrez la chaine ADN: atgcaa
Entrez la chaine sequence: ca
Il y a 33.333333 % de ca dans votre chaine d ADN qui contient 6 caracteres
```

----- { *prendre une nouvelle copie* } -----

### 3<sup>ème</sup> partie : tableaux (7 points)

#### 1. Pair ou impair ?

Écrire un algorithme permettant de remplir un tableau avec une série de `n` nombres entiers entrés au clavier, puis de calculer la moyenne des nombres pairs de ce tableau. Le nombre `n` sera lui-aussi saisi au clavier.

L'algorithme affichera "Ce tableau est pair !", si tous les nombres impairs sont inférieurs à cette moyenne et "Ce tableau est impair !" sinon.

Écrire le programme Python correspondant.

#### 2. Les pairs d'abord !

- a) Écrire une fonction python `classif_pair_impair(tab)` qui prend en entrée un tableau `tab` de nombres entiers et classe le tableau de manière à ce que tous les nombres pairs se retrouvent au début du tableau et les nombres impairs à la fin. L'ordre des nombres pairs dans le tableau sera gardé. Le tableau sera modifié sans avoir recours à un autre tableau.
- b) Écrire un programme principal qui fait appel à la fonction `classif_pair_impair(tab)` et affiche le résultat comme dans l'exemple suivant :

Exemple :

```
Tableau transmis à la fonction : [24, 13, 17, 16, 2, 5, 7, 42]
```

```
Tableau après classification pairs et impairs : [24, 16, 2, 42, 13, 17, 5, 7]
```

- c) La fonction `classif_pair_impair` modifie le tableau passé en paramètre, ce qui n'est pas une bonne pratique en général. En utilisant cette fois plusieurs tableaux, écrire une fonction `classif_pair_impair_2` qui retourne un tableau classé comme précédemment mais sans modifier le tableau passé en paramètre.