

Dans les énoncés, on précise si on demande une **fonction** ou un **programme**, faites attention à bien respecter l'énoncé.

Exercice I.1 (*Manipulation de variables (1 point)*)

Donnez les valeurs de chaque variable à la fin de l'exécution du programme ci-contre. Quel message s'affiche ?

```
i,j=65,6.5
x,k=30,i/j
if (k*3==x and x%3==0):
    print("Blue")
else:
    print("Red")
```

Exercice I.2 (*Exécution de fonction (2 points)*)

On considère le programme ci-contre, expliquez précisément son comportement en détaillant les appels de fonction réalisés, les valeurs retournées par ces appels, et la valeur finale affichée.

```
def first(n):
    if (n%2==0):
        return 2*n
    else:
        return 3*n-1

def second(L):
    for i in range(len(L)):
        L[i]=first(L[i])
    return L

print(second([3,0,2]))
```

Exercice I.3 (*Fonction et suites (5 points : 1+1+1+2)*)

On considère la fonction f dépendant d'une variable $x \in \mathbb{Z}$ définie par :

$$\forall x \in \mathbb{Z}, f(x) = \begin{cases} x^2 - 35x + 1 & \text{si } x \text{ est pair et } x > 35 \\ -x^2 + 3x + 2 & \text{si } x \text{ est impair et } x \leq 11 \\ x^3 - x^2 + x - 1 & \text{sinon.} \end{cases}$$

On considère ensuite la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 10$ et $u_{n+1} = f(u_n)$ pour tout $n \in \mathbb{N}$.

1. Écrivez une **fonction fonction_f** qui prend en entrée un entier x et retourne la valeur de $f(x)$.
2. En utilisant cette fonction, écrivez un **programme** qui demande à l'utilisateur de saisir un entier x , puis qui affiche le message suivant :
"La valeur de la fonction f pour x=35 est 41684"
(ici 35 est la valeur saisie par l'utilisateur).
3. En utilisant la fonction définie à la question 1, écrivez une **fonction récursive suite_u** prenant en entrée un entier naturel n et retournant la valeur de u_n .
4. Écrivez une **fonction non récursive indice_max** prenant en entrée un entier naturel n , et retournant l'indice du plus grand des termes $u_0, u_1, u_2, \dots, u_n$.

Exercice I.4 (*Listes (7 points : 1+2+2+1+1)*)

Vous allez écrire des fonctions manipulant des listes d'entiers relatifs (vous ne vérifierez pas que c'est effectivement le cas). Attention, vous ne devez pas utiliser les fonctions prédéfinies pour la recherche d'éléments dans une liste.

1. Écrire une **fonction present** qui prend en argument une liste L et un entier x et qui retourne `True` si x est présent dans L , et `False` sinon.
2. Écrire une **fonction compte_sup** qui prend en argument une liste L et un entier x et qui retourne le nombre d'éléments de L qui sont supérieurs ou égaux à x . Écrire également une **fonction compte_inf** qui prend en argument L et x et qui retourne le nombre d'éléments de L qui sont inférieurs ou égaux à x .
3. Un élément x d'un ensemble de N valeurs est appelé une médiane si x vérifie les deux propriétés suivantes :
 - le nombre d'éléments supérieurs ou égaux à x est au moins $N/2$
 - le nombre d'éléments inférieurs ou égaux à x est au moins $N/2$On peut démontrer que tout ensemble possède une médiane (parfois plusieurs).
Écrire une **fonction mediane** qui prend en argument une liste L et qui retourne une médiane de L .
4. Étant donné un vecteur $\vec{x} = (x_i)_{1 \leq i \leq n}$, la norme 1 de \vec{x} est définie comme $\|\vec{x}\|_1 = \sum_{i=1}^n |x_i|$.
Écrire une **fonction norme1** qui prend en argument une liste L et qui retourne la norme 1 du vecteur représenté par L . Vous pouvez utiliser la fonction `abs` existant par défaut en Python et donnant la valeur absolue.
5. Étant donné un vecteur $\vec{x} = (x_i)_{1 \leq i \leq n}$, la norme 2 de \vec{x} est définie comme $\|\vec{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$.
Écrire une **fonction norme2** qui prend en argument une liste L et qui retourne la norme 2 du vecteur représenté par L .

Exercice I.5 (*Chaînes de caractères (5 points : 1+2+2)*)

On va chercher dans cet exercice à écrire des fonctions permettant de corriger une chaîne de caractères. Attention, vous ne devez pas utiliser les fonctions ou méthodes prédéfinies, et vous devez donc construire les nouvelles chaînes à l'aide de boucles.

1. On suppose que la chaîne de caractères a été tapée sur un clavier défectueux, sur lequel les lettres 'a' et 'q' sont inversées.
Écrire une **fonction echange** qui prend en argument une chaîne de caractères `chaîne` et qui retourne la chaîne de caractères obtenue en remplaçant les occurrences du caractère 'a' par le caractère 'q', et réciproquement, les occurrences de 'q' par 'a'.
Exemple : `echange('‘aui est lq?’')` retourne `‘‘qui est la?’’`
2. On suppose maintenant que la personne qui a tapé le texte a une certaine dyslexie, et qu'à chaque fois qu'elle doit taper la séquence 'ea', elle a tapé 'ae' à la place.
Écrire une **fonction corrige_dys** qui prend en argument une chaîne de caractères `chaîne` et qui retourne la chaîne de caractères obtenue en corrigeant les défauts présentés précédemment. Plus précisément, à chaque fois que `chaîne` contient 'ae', on remplace ces deux caractères par 'ea'.
Exemple : `corrige_dys('‘tu es baeu?’')` retourne `‘‘tu es beau?’’`
3. Pour finir, on suppose que la personne qui a saisi le texte a la main un peu lourde, et qu'il lui arrive d'appuyer trop longtemps sur une touche. Or, en français, une lettre n'est jamais répétée plus de deux fois.
Écrire une **fonction corrige_rep** qui prend en argument une chaîne de caractères `chaîne` et qui retourne la chaîne de caractères obtenue en corrigeant les défauts présentés précédemment. Plus précisément, à chaque fois qu'un caractère apparaît plus de deux fois consécutivement, alors on élimine les répétitions pour en laisser exactement deux.
Exemple : `corrige_rep('‘l'effffort est immense?’')` retourne `‘‘l'effort est immense?’’`