

Devoir d'informatique

Consignes

- Le devoir se fera sur copie double uniquement.
- Le numéro de chaque exercice et de chaque question devra être indiqué sur votre copie.
- Les indentations devront correctement figurer sur votre copie. Vous pourrez par exemple tracer une barre verticale.
- Pour chaque fonction vous donnerez au plus une ligne de commentaire permettant de spécifier votre fonction.

Exercice 1 – Structure If

Question 1 Implémenter la fonction `f_conditionnel(x:int)-> int` correspondant à la fonction mathématique suivante :

$$f : x \rightarrow \begin{cases} 2 & \text{si } x < 0 \\ 10 & \text{si } x = 0 \\ -x + 2 & \text{si } x \in]0; 3] \\ -1 & \text{si } x > 3 \end{cases}$$

Exercice 2 – Structure for

Question 2 Ecrire une fonction `multiples_3(n:int) ->None` affichant de tous les multiples de 3 compris entre 0 et n inclus. On utilisera une boucle `for`.

Exercice 3 – Structure while

Question 3 Ecrire une fonction `multiples_3w(n:int) ->None` affichant de tous les multiples de 3 compris entre 0 et n . On utilisera une boucle `while`.

Exercice 4 – Structure for ou while

Question 4 Ecrire une fonction `somme_pairs(n:int)->n` calculant la somme des entiers pairs de 1 à n exclus.

Exercice 5 – Implémentation d'une suite

On pose $u_0 = 1$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = \frac{1}{2} \left(u_n + \frac{n+1}{u_n} \right)$.

Question 5 Ecrire une fonction `u(n:int) ->n` permettant de renvoyer le n ème terme de la suite u .

Exercice 6 –

On donne la fonction suivante :

```
def inv(n):
    """Somme des inverses des n premiers
    entiers naturels non nuls"""
    s = 0
    for k in range(n):
        x = 1/k
    s = s+x
    return s
```

Question 6 Donner l'évolution des variables k et x en fin d'itération lors de l'appel suivant `inv(4)`. Quelle est la valeur renvoyée par la fonction?

Question 7 Corriger la fonction pour qu'elle réponde à l'objectif annoncé.

Exercice 7 – ADN

Un brin d'ADN est composé de bases nucléiques notées 'a', 'c', 'g' ou 't'.

Question 8 Ecrire une fonction `verifie(adn:str)->bool` qui renvoie `True` si la chaîne ne contient que des bases nucléiques 'a', 'c', 'g' ou 't'. Sinon, la fonction renvoie `False`.

Question 9 Ecrire une fonction `ajout(adn:str, base:str)->str` qui ajoute les bases `base` à l'`adn` et qui renvoie la chaîne résultante. Ainsi `ajout('acgt', 'gt')` renvoie `'acgtgt'`.

Question 10 Ecrire une fonction `compte(adn:str, base:str)->int` qui renvoie le nombre d'occurrences de la base. La fonction renverra `-1` si la chaîne n'est pas valide.

Question 11 Ecrire une fonction `proportion(adn:str, base:str)->int` qui renvoie la proportion de la base `base` dans le brin d'`adn`. La fonction renverra `0` si la chaîne n'est pas valide.