**Question 1**

**Texte de la question**

Que retourne la fonction suivante ? (L désigne une liste de réels.)

def algo1(L):

    n = len(L)

    S = 0

    for k in range(n):

        S = S + L[k]

    return S

**Réponse :**

**Question 2**

**Texte de la question**

Que retourne la fonction suivante ? (L désigne une liste de réels.)

def algo2(L):

    n = len(L)

    M = L[0]

    for k in range(1,n):

        if L[k]>M:

            M = L[k]

    return M

**Réponse :**

**Question 3**

**Texte de la question**

Que retourne la fonction suivante ? (n désigne un entier naturel.)

def algo3(n):

    U = 5

    k = 0

    while k<n:

        U = 2\*U+6/(k+1)

        k = k+1

    return U

**Réponse :**

**Question 4**

**Texte de la question**

Que retourne la fonction suivante ? (n désigne un entier naturel.)

def algo4(n):

    U = 5

    for k in range(0,n):

        U = 2\*U+6/(k+1)

    return U

**Réponse :**

**Question 5**

**Texte de la question**

Écrire une fonction **positionDuMax(L)** qui, pour une liste **L**, donne la/une position du plus grand des éléments de la liste.

Par exemple : **positionDuMax([5,2,-3,5,-9])** doit retourner soit 0 soit 3, au choix.

**Réponse :**

**Question 6**

**Texte de la question**

Écrire une fonction **depasse(M)** qui donne le plus petit indice **n** pour lequel **un ≥ M**, où **M** est un réel donné et la suite est définie par **u0=3** et **un+1 = un(un-1)**.

(Comme la suite diverge vers +∞, on peut toujours trouver un tel n.)

**Réponse :**

**Question 7**

**Texte de la question**

Écrire une fonction sommeTermesPositifs(L) qui, pour une liste L, donne la somme des éléments positifs de la liste.

Par exemple : sommeTermesPositifs([2,-3,5,-9]) doit retourner 7.

**Réponse :**

**Question 8**

**Texte de la question**

Écrire une fonction **sommeU(n)** qui, pour un entier naturel **n**, donne la valeur de la somme **u0+u1+...+un**.

Où **u** est la suite est définie par **u0=3** et **un+1 = un(un-1)**.

**Réponse :**