

Devoir surveillé 1

Durée : 1H. Calculatrices interdites. **Les candidats sont invités à porter une attention particulière à la rédaction : les copies illisibles ou mal présentées seront pénalisées.**

Lisez l'énoncé **attentivement** avant de répondre à chaque question.

Exercice 1

Donner la décomposition en binaire de 213.

Exercice 2

Un CAN (convertisseur analogique-numérique, qui transforme un signal physique en une suite de bit), dans une situation pratique non définie ici, doit pouvoir distinguer entre 5000 valeurs de la quantité mesurée (il y a potentiellement 5000 valeurs différentes que l'on souhaite pouvoir obtenir). ON considère que ces valeurs peuvent être représentées par les entiers de 0 à 4999.

Quelle profondeur d'échantillonnage doit-il utiliser ? (ou encore, combien de bits, au minimum, doit-on utiliser pour pouvoir stocker l'une des mesures ?)

Exercice 3

On considère la fonction suivante :

```

1 def expo(x, n):
2     res = 1
3     j = n
4     while j > 0:
5         res = res * x
6         j = j - 1
7     return res

```

1. A l'aide d'un tableau dont les colonnes seront `res` et `j`, expliciter les valeurs prises par ces variables lors du calcul de `expo(2, 3)`.
2. Dans le cas général, quelles sont les valeurs prises par `j` et `res` lors de l'exécution de `expo(x, n)` ?
3. On appelle ici opération élémentaire : une opération sur deux nombres (+, -, *, /) ou une affectation d'une valeur à une variable.

On note $C(n)$ le nombre d'opérations élémentaires nécessaires au calcul de x^n par cette fonction. Montrer que $C(n) = 4n + 2$.

Exercice 4

Écrire une fonction **factorielle(n)** qui prend comme argument un entier positif n (ne pas le vérifier dans la fonction) et retourne $n! = 1 \times 2 \times \dots \times n$.

Exercice 5

Écrire une fonction **occurrences(s, c)** qui prend comme argument une chaîne de caractère s et un caractère c et retourne le nombre de fois où le caractère c est présent dans la chaîne. Expliquer rapidement le rôle de chaque variable locale.

Exercice 6

Écrire une fonction **maxi(L)** dont l'argument est une liste de nombres L et qui retourne la valeur du plus grand nombre contenu dans L .

Exercice 7

On considère la fonction suivante, où L est une liste de nombres entiers

```

1 def f(L):
2     s = 1
3     for i in range(len(L)):
4         for j in range(i + 1, len(L)):
5             if L[j] < L[i]:
6                 s = -s
7     return s

```

1. Que vaut $f([1, 3, 2])$? On pourra, comme pour l'exercice 1 considérer un tableau représentant l'évolution des variables lors de l'exécution.
2. Quelles sont les valeurs de retour possibles pour f ?
3. On note n la longueur de la liste L . Exprimer en fonction de n le nombre de fois où l'instruction `if` est exécutée.
4. On suppose que L est triée dans l'ordre croissant. Que vaut $f(L)$? Justifier !
5. Donner un exemple simple de liste L non triée pour laquelle $f(L)$ vaut 1.