

Devoir 3 – Complément à 2, programmation Python

Exercice 1 : 4 points

Cet exercice est un QCM. Pour chaque question, une seule des réponses proposées est correcte. Une mauvaise réponse, l'absence de réponse ou choisir plusieurs réponses ne rapporte, ni n'enlève aucun point. Écrire sur votre copie le numéro de la question ainsi que la réponse choisie.

1. On considère les instructions suivantes :

```
ch = "Donovan"  
c = ch[4]
```

Que contient la variable c ?

- (a) "D" (b) "o" (c) "n" (d) "v"

2. On considère l'instruction `liste_mystere = [2 + i for i in range(5)]`. Que contient la variable `liste_mystere` ?

- (a) [2, 3, 4, 5] (c) [2, 3, 4, 5, 6, 7]
(b) [2, 3, 4] (d) [2, 3, 4, 5, 6]

3. On considère les instructions suivantes :

```
ch = "J'aime les tests de NSI."  
c = ch[-1]
```

Que contient la variable c ?

- (a) "J" (c) "I"
(b) "." (d) Rien, le code donné produit une erreur.

4. On considère les instructions suivantes :

```
prenom = ["Aika", "Marie", "Vaitiare", "Mana"]  
p = prenom[1][3]
```

Que contient la variable p ?

- (a) "M" (b) "r" (c) "k" (d) "v"

Exercice 2 : 3 points

On utilise la méthode du complément à 2 sur un octet :

1. Représenter 49.
2. Représenter -73.
3. Quel entier est représenté par $\overline{1011\ 1000}^2$?

Exercice 3 : 3 points

On souhaite écrire une fonction `liste_aleatoire(n, a, b)` qui renvoie une liste de taille n contenant des entiers tirés au hasard entre deux entiers a et b (ces deux entiers étant inclus).

1. Donner un exemple de liste renvoyée par l'appel `liste_aleatoire(6, 3, 20)`.
2. Écrire la fonction `liste_aleatoire(n, a, b)`.

Exercice 4 : 3 points

On souhaite écrire une fonction `occurrences(v, L)` qui renvoie le nombre d'occurrences de la valeur `v` dans la liste `L`.

1. Proposer trois tests pour la fonction `occurrences(v, L)` en utilisant `assert`.
2. Écrire la fonction `occurrences`.

Exercice 5 : 3 points

On modélise la représentation binaire d'un entier positif ou nul par un tableau d'entiers dont les éléments sont 0 ou 1. Par exemple, le tableau `[1, 0, 1, 0, 0, 1, 1]` représente l'écriture binaire de l'entier dont l'écriture décimale est :

$$2^{**6} + 2^{**4} + 2^{**1} + 2^{**0} = 83.$$

Écrire la fonction `convertir(T)` répondant aux spécifications suivantes :

```
def convertir(T):  
    """  
    T est un tableau d'entiers, dont les éléments sont 0 ou 1,  
    représentant un entier écrit en binaire.  
    Renvoie l'écriture décimale de l'entier positif dont la  
    représentation binaire est donnée par T.  
    """
```

Quelques exemples :

- `convertir([1, 0, 1, 0, 0, 1, 1])` renvoie 83;
- `convertir([1, 0, 0, 0, 0, 0, 1, 0])` renvoie 130.

Exercice 6 : 4 points

Écrire une fonction `recherche(L, n)` qui prend en paramètre une liste non vide `L` d'entiers et un entier `n`, et qui renvoie l'indice de la dernière occurrence de l'élément cherché. Si l'élément `n` n'est pas présent, la fonction renvoie la longueur de la liste.

Quelques exemples :

- `recherche([5, 3], 1)` renvoie 2;
- `recherche([2, 4], 2)` renvoie 0;
- `recherche([2, 3, 5, 2, 4], 2)` renvoie 3.